

NCE/12/00056 — Apresentação do pedido - Novo ciclo de estudos

Apresentação do pedido

Perguntas A1 a A4

A1. Instituição de Ensino Superior / Entidade Instituidora:

Universidade Nova De Lisboa

A1.a. Outras Instituições de Ensino Superior / Entidades Instituidoras:

A2. Unidade(s) orgânica(s) (faculdade, escola, instituto, etc.):

Faculdade De Ciências E Tecnologia (UNL)

A3. Designação do ciclo de estudos:

Engenharia Informática

A3. Study cycle name:

Computer Science and Engineering

A4. Grau:

Mestre (M)

Perguntas A5 a A10

A5. Área científica predominante do ciclo de estudos:

Informática

A5. Main scientific area of the study cycle:

Informatics

A6.1. Classificação da área principal do ciclo de estudos (3 algarismos), de acordo com a Portaria n.º 256/2005, de 16 de Março (CNAEF):

523

A6.2. Classificação da área secundária do ciclo de estudos (3 algarismos), de acordo com a Portaria n.º 256/2005, de 16 de Março (CNAEF), se aplicável:

481

A6.3. Classificação de outra área secundária do ciclo de estudos (3 algarismos), de acordo com a Portaria n.º 256/2005, de 16 de Março (CNAEF), se aplicável:

<sem resposta>

A7. Número de créditos ECTS necessário à obtenção do grau:

300

A8. Duração do ciclo de estudos (art.º 3 DL-74/2006, de 26 de Março):

5 anos (10 semestres)

A8. Duration of the study cycle (art.º 3 DL-74/2006, March 26th):
5 years (10 semesters)

A9. Número de vagas proposto:
150

A10. Condições de acesso e ingresso:
Provas de Ingresso para 2013/2014:

Um dos seguintes conjuntos:
(19 Matemática A) OU (04 Economia, 19 Matemática A) OU (07 Física e Química, 19 Matemática A)

Nota de candidatura: 95 pontos
Prova de ingresso: 95 pontos

Fórmula de Cálculo:
Média do Ensino Secundário: 60%
Provas de ingresso: 40%

A10. Entry Requirements:
Admission Examinations 2013/2014:

One of the following sets:
(19 Mathematics A) OR (04 Economics, 19 Mathematics A) OR (07 Physics and Chemistry, 19 Mathematics A)

Application mark: 95 / 200
Admission examination: 95 / 200

Computation Rule:
Secondary School Grade Average: 60%
Admission examinations: 40%

Pergunta A11

Pergunta A11

A11. Ramos, opções, perfis, maior/menor ou outras formas de organização de percursos alternativos em que o ciclo de estudos se estrutura (se aplicável):
Não

A11.1. Ramos, opções, perfis, maior/menor, ... (se aplicável)

A11.1. Ramos, opções, perfis, maior/menor, ou outras formas de organização de percursos alternativos em que o ciclo de estudos se estrutura (se aplicável) / Branches options, profiles, major/minor, or other forms of organization of alternative paths compatible with the structure of the study cycle (if applicable)

Ramos/Opções/... (se aplicável):

Branches/Options/... (if applicable):

<sem resposta>

A12. Estrutura curricular

Mapa I -

A12.1. Ciclo de Estudos:
Engenharia Informática

A12.1. Study Cycle:*Computer Science and Engineering***A12.2. Grau:***Mestre (MI)***A12.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):**

<sem resposta>

A12.3. Branches, options, profiles, major/minor, or other forms (if applicable):

<no answer>

A12.4. Áreas científicas e créditos que devem ser reunidos para a obtenção do grau / Scientific areas and credits that must be obtained for the awarding of the degree

Área Científica / Scientific Area	Sigla / Acronym	ECTS Obrigatórios / Mandatory ECTS	ECTS Optativos* / Optional ECTS*
Informática / Informatics	I	153	75
Matemática / Mathematics	M	36	0
Física / Physics	F	6	0
Engenharia Eletrotécnica e Computadores / Electrical and Computer Engineering	EEC	6	0
Ciências Humanas e Sociais / Social Sciences and Humanities	CHS	6	0
Competências Complementares / Transferable Skills	CC	6	0
Qualquer área científica / Any Scientific Area	QAC	0	12
(7 Items)		213	87

Perguntas A13 e A14**A13. Regime de funcionamento:***Diurno***A13.1. Se outro, especifique:**

<sem resposta>

A13.1. If other, specify:

<no answer>

A14. Observações:

A FCT/UNL é pioneira no ensino da Engenharia Informática em Portugal. A presente proposta assenta na experiência de ensino acumulada em 37 anos pelo seu Departamento de Informática (DI), complementada com a opinião de consultores externos, e consubstancia uma visão atualizada sobre a formação do Engenheiro Informático de conceção.

O ciclo de estudos (curso) proposto apresenta a estrutura de 2 ciclos integrados, compatível com os princípios de Bolonha, visando a formação em 5 anos de engenheiros com o grau de mestre (Mestrado Integrado). A extensão do curso é a julgada adequada para a concessão de formação e maturação compatíveis com a prática profissional da Engenharia de Conceção, alinhada com o estabelecido na escola para as Engenharias, com os pressupostos da acreditação da Ordem dos Engenheiros, e com programas afins em várias escolas de referência. Este objetivo de formação é assegurado sem impossibilitar a mobilidade, sendo que, mesmo não sendo o seu alvo, o curso concede a eventuais diplomados de 1º ciclo uma formação de base em Informática coerente e relevante.

O curso completo estrutura-se em 3 partes, de acordo com as recomendações do ACM/IEEE CS Curriculum 2013: o bloco base, o bloco de consolidação e o bloco de especialização, abrangendo o 1º ciclo todo o bloco base e parte do bloco de consolidação.

O bloco base (168 ECTS, 56% do curso) consiste em unidades curriculares (UC) obrigatórias (1º e 2º anos, mais 6 UC no 3º ano) que cobrem (e excedem) o nível 1 ACM/IEEE. Entre o 3º e o 4º ano é oferecida a UC "Atividade Prática de Desenvolvimento Curricular", de 15 ECTS, que concede experiência de projeto, de iniciação à investigação, ou outra atividade coerente, em contexto variável (laboratório, empresa, outro). Esta UC enquadra-se no "Perfil Curricular FCT", marca diferenciadora da escola, suportada em UC de competências transversais, imersão em atividades de realização prática e multidisciplinaridade.

O bloco de consolidação (60 ECTS, 20% do curso), inclui 11 UC de Informática, que cobrem matérias tipicamente classificadas no nível 2 ACM/IEEE. Pretende-se com estas UC, oferecidas no 3º-4º ano, conceder uma formação muito sólida, de banda larga, em Engenharia Informática. De forma a permitir alguma flexibilidade, sem comprometer o requisito de abrangência, os alunos terão que obter aprovação em 7 de entre as 11 UC oferecidas.

Ao bloco de consolidação segue-se o bloco de especialização (30 ECTS, 10% do curso), constituído por UC opcionais, que cobrem matérias do nível 3 ACM/IEEE, permitindo ao estudante aprofundar conhecimentos numa subárea da Engenharia Informática. Os tópicos cobertos no bloco de especialização, e, em certa medida, no bloco de consolidação, complementam temas clássicos com tendências mais atuais, de acordo com as recomendações ACM/IEEE 2013, assegurando o alinhamento do curso com as necessidades da sociedade, e com a investigação atual na área. O curso culmina com a Dissertação de Mestrado (42 ECTS, 14% do curso).

A14. Observations:

FCT/UNL pioneered Computer Science and Engineering (CSE) higher education in Portugal, with the Informatics Department founded 37 years ago. This proposal builds on the accumulated experience of D/FCT/UNL, and offers an updated vision on the educational path of a modern professional engineer in the CSE field.

The proposed program structure, composed by two integrated cycles, targets master level education after 5 years of study, duration considered adequate for building up, we believe, the level of maturation compatible with the professional practice of conception-oriented engineering, in alignment with the established practice at FCT/UNL for engineering programs, with the accreditation procedures of "Ordem dos Engenheiros" in Portugal, and with many similar programs in reference schools. This goal is ensured without hindering student mobility (in and out) at the interface between 1st and 2nd cycle. The adopted format provides 1st cycle graduates, even if not our main target, some coherent and relevant basic education in CSE.

The program is composed by three parts, following the more recent recommendations of the ACM/IEEE Computing Curricula 2013: a core tier, a consolidation tier, and a specialization tier.

The core tier (around 56% of the curriculum - 168 ECTS), is composed by mandatory curricular units (1st and 2nd year, plus 6 units in 3th year). By the end of the core tier the unit "Practical Activity of Curricular Development" (15 ECTS) offers to students an hands-on experience, project or introductory research activity, in a "real" setting out of the classroom context. This unit also fits in the so-called "FCT Curricular Profile", common to all programs in our school. This profile aims to clearly mark a distinguishing character resulting units such as Soft Skills for Science and Technology (1st year), Science, Technology, and Society (2nd year), the Undergraduate Research / Placement Opportunities Program (3rd year) and Entrepreneurship (4th year).

The consolidation tier spans around 20% of the curriculum (60 ECTS) and includes a set of 11 CU (units), covering subject matter typically classified at tier 2 by ACM/IEEE. These CU, placed in 3th-4th year provide a solid education, and in breath advanced conceptual design skills in CSE topics. To allow flexibility, with compromising a breath requirement, students are required to complete 7 among the 11 offered CU.

The specialization tier (30 ECTS, 10%) follows the consolidation tier and offers several elective CU covering diverse subjects mostly from ACM/IEEE tier 3, and may allow the student to deepen his knowledge and skills in some subarea of CSE. Topics covered in the specialization tier (and to some extent in the consolidation tier) complement

classical themes with modern trends, according to the ACM/IEEE 2013 recommendations, aligned with the needs of society, and current scientific research directions. The curriculum concludes with the written Master Dissertation (42 ECTS, 14%).

Instrução do pedido

1. Formalização do pedido

1.1. Deliberações

Mapa II - Aprovação pelo Reitor da UNL, ouvido o Colégio de Diretores

1.1.1. Órgão ouvido:

Aprovação pelo Reitor da UNL, ouvido o Colégio de Diretores

- 1.1.2. Cópia de acta (ou extrato de acta) ou deliberação deste órgão assinada e datada (PDF, máx. 100kB):
[1.1.2._Despacho Senhor Reitor_FCT_MI Engenharia Informática_10-10-2012.pdf](#)

Mapa II - Conselho Científico da FCT - UNL

- 1.1.1. Órgão ouvido:

Conselho Científico da FCT - UNL

- 1.1.2. Cópia de acta (ou extrato de acta) ou deliberação deste órgão assinada e datada (PDF, máx. 100kB):
[1.1.2._Declaração do CC _ MIEI.pdf](#)

Mapa II - Conselho Pedagógico da FCT - UNL

- 1.1.1. Órgão ouvido:

Conselho Pedagógico da FCT - UNL

- 1.1.2. Cópia de acta (ou extrato de acta) ou deliberação deste órgão assinada e datada (PDF, máx. 100kB):
[1.1.2._Declaração_CP_MIEI.pdf](#)

1.2. Docente(s) responsável(eis)

- 1.2. Docente(s) responsável(eis) pela coordenação da implementação do ciclo de estudos
A(s) respectiva(s) ficha(s) curricular(es) deve(m) ser apresentada(s) no Mapa V.

Luís Manuel Marques da Costa Caires

2. Plano de estudos

Mapa III - - 1º Ano / 1º Semestre

- 2.1. Ciclo de Estudos:

Engenharia Informática

- 2.1. Study Cycle:

Computer Science and Engineering

- 2.2. Grau:

Mestre (M)

- 2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):

<sem resposta>

- 2.3. Branches, options, profiles, major/minor, or other forms (if applicable):

<no answer>

- 2.4. Ano/semestre/trimestre curricular:

1º Ano / 1º Semestre

- 2.4. Curricular year/semester/trimester:

1st Year / 1st Semester

2.5. Plano de Estudos / Study plan

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS / Observações Observations
--	---	------------------------------	--	--	---------------------------------------

						(5)
Análise Matemática IE / Mathematical Analysis IE	M	Semestral / Semester	168	T:42 P:42	6	Obrigatória / Mandatory
Álgebra Linear e Geometria Analítica / Linear Algebra and Analytic Geometry	M	Semestral / Semester	168	T:42 P:28	6	Obrigatória / Mandatory
Introdução à Programação / Introductory Programming	I	Semestral / Semester	250	TP:63 P:21	9	Obrigatória / Mandatory
Sistemas Lógicos / Digital Systems	EEC	Semestral / Semester	163	T:42 P:42 OT:2	6	Obrigatória / Mandatory
Competências Transversais para Ciência e Tecnologia / Soft Skills for Science and Technology	CC	Semestral / Semester	80	TP:10 P:50	3	Obrigatória / Mandatory

(5 Items)

Mapa III - - 1º Ano / 2º Semestre

2.1. Ciclo de Estudos:

Engenharia Informática

2.1. Study Cycle:

Computer Science and Engineering

2.2. Grau:

Mestre (M)

2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):

<sem resposta>

2.3. Branches, options, profiles, major/minor, or other forms (if applicable):

<no answer>

2.4. Ano/semestre/trimestre curricular:

1º Ano / 2º Semestre

2.4. Curricular year/semester/trimester:

1st Year / 2nd Semester

2.5. Plano de Estudos / Study plan

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Análise Matemática IIE / Mathematical Analysis IIE	M	Semestral / Semester	168	T:42 P:42	6	Obrigatória / Mandatory
Matemática Discreta / Discrete Mathematics	M	Semestral / Semester	168	T:42 P:28 OT:3	6	Obrigatória / Mandatory
Arquitetura de Computadores / Computer Architecture	I	Semestral / Semester	250	T:42 P:28 OT:2	9	Obrigatória / Mandatory
Programação Orientada pelos Objetos / Object Oriented Programming	I	Semestral / Semester	250	T:42 P:28	9	Obrigatória / Mandatory

(4 Items)

Mapa III - - 2º Ano / 3º Semestre**2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (MI)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):***<sem resposta>***2.3. Branches, options, profiles, major/minor, or other forms (if applicable):***<no answer>***2.4. Ano/semestre/trimestre curricular:***2º Ano / 3º Semestre***2.4. Curricular year/semester/trimester:***2nd Year / 3th Semester***2.5. Plano de Estudos / Study plan**

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Algoritmos e Estruturas de Dados / Algorithms and Data Structures	I	Semestral / Semester	250	T:42 P:28 OT:3	9	Obrigatória / Mandatory
Fundamentos de Sistemas de Operação / Operating Systems Foundation	I	Semestral / Semester	250	T:42 P:28 OT:2	9	Obrigatória / Mandatory
Lógica Computacional / Computational Logic	I	Semestral / Semester	168	T:28 P:42	6	Obrigatória / Mandatory
Física / Physics	F	Semestral / Semester	168	T:42 P:14	6	Obrigatória / Mandatory

(4 Items)**Mapa III - - 2º Ano / 4º Semestre****2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (MI)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):***<sem resposta>*

2.3. Branches, options, profiles, major/minor, or other forms (if applicable):

<no answer>

2.4. Ano/semestre/trimestre curricular:

2º Ano / 4º Semestre

2.4. Curricular year/semester/trimester:

2nd Year / 4th Semester

2.5. Plano de Estudos / Study plan

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS / (5)	Observações / Observations (5)
Probabilidades e Estatística E / Probability and Statistics E	M	Semestral / Semester	168	TP:56	6	Obrigatória / Mandatory
Teoria da Computação / Theoretical Computer Science	I	Semestral / Semester	168	T:42 P:28 OT:3	6	Obrigatória / Mandatory
Bases de Dados / Databases	I	Semestral / Semester	250	T:42 P:28	9	Obrigatória / Mandatory
Linguagens e Ambientes de Programação / Programming Languages and Environments	I	Semestral / Semester	164	T:42 P:28 OT:3	6	Obrigatória / Mandatory
Ciência, Tecnologia e Sociedade / Science, Technology and Society	CHS	Semestral / Semester	80	TP:32 S:8	3	Obrigatória / Mandatory

(5 Items)

Mapa III - - 3º Ano / 5º Semestre**2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (M)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):**

<sem resposta>

2.3. Branches, options, profiles, major/minor, or other forms (if applicable):

<no answer>

2.4. Ano/semestre/trimestre curricular:

3º Ano / 5º Semestre

2.4. Curricular year/semester/trimester:

3th Year / 5th Semester

2.5. Plano de Estudos / Study plan**Observações**

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS / Observations (5)
Computação Gráfica e Interfaces / Computer Graphics and Interfaces	I	Semestral / Semester	162	T:42 P:28 OT:2	6 Obrigatória / Mandatory
Inteligência Artificial / Artificial Intelligence	I	Semestral / Semester	168	T:42 P:28 OT:1	6 Obrigatória / Mandatory
Métodos de Desenvolvimento de Software / Software Development Methods	I	Semestral / Semester	168	T:42 P:28	6 Obrigatória / Mandatory
Redes de Computadores / Computer Networks	I	Semestral / Semester	165	T:42 P:28	6 Obrigatória / Mandatory
Unidade Curricular do Bloco Livre B / Unrestricted Elective B	QAC	Semestral / Semester	165	depende da UC escolhida/dependent of choice	6 Optativa / Elective

(5 Items)

Mapa III - - 3º Ano / 6º Semestre**2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (M)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):***<sem resposta>***2.3. Branches, options, profiles, major/minor, or other forms (if applicable):***<no answer>***2.4. Ano/semestre/trimestre curricular:***3º Ano / 6º Semestre***2.4. Curricular year/semester/trimester:***3th Year / 6th Semester***2.5. Plano de Estudos / Study plan**

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS / Observations (5)
Atividade Prática de Desenvolvimento Curricular / Practical Activity for Curricular Development	I	Semestral / Semester	420	P:28 OT:10 S:14	15 Obrigatória / Mandatory
Programa de Oportunidades / Opportunities Program	I	Semestral / Semester	80	OT:7	3 Optativa / Elective
Unidade de Consolidação I / Consolidation Unit I	I	Semestral / Semester	168	depende da UC escolhida/dependent of choice	6 Optativa / Elective
Unidade de Consolidação II /		Semestral		depende da UC	Optativa /

Consolidation Unit II

I

/ Semester 168

escolhida/dependent
of choice 6

Elective

(4 Items)**Mapa III - - 3º Ano / 6º Semestre - Opções - Programa de Oportunidades****2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (MI)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):***<sem resposta>***2.3. Branches, options, profiles, major/minor, or other forms (if applicable):***<no answer>***2.4. Ano/semestre/trimestre curricular:***3º Ano / 6º Semestre - Opções - Programa de Oportunidades***2.4. Curricular year/semester/trimester:***3th Year / 6th Semester - Electives - Opportunities Program***2.5. Plano de Estudos / Study plan**

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Programa de Introdução à Prática Profissional / Undergraduate Practice Opportunities Program	I	Semestral / Semester	80	OT:7	3	Optativa / Elective
Programa de Introdução à Investigação Científica / Undergraduate Research Opportunities Program	I	Semestral / Semester	80	OT:7	3	Optativa / Elective

(2 Items)**Mapa III - - 4º Ano / 7º Semestre****2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (MI)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):**

<sem resposta>

2.3. Branches, options, profiles, major/minor, or other forms (if applicable):

<no answer>

2.4. Ano/semestre/trimestre curricular:

4º Ano / 7º Semestre

2.4. Curricular year/semester/trimester:

4th Year / 7th Semester

2.5. Plano de Estudos / Study plan

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS / Observations (5)	Observações / Observations (5)
Introdução à Investigação Operacional / Introduction to Operational Research	M	Semestral / Semester	168	T:28 P: 28	6	Obrigatória / Mandatory
Unidade de Consolidação III / Consolidation Unit III	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Unidade de Consolidação IV / Consolidation Unit IV	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Unidade de Consolidação V / Consolidation Unit V	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Unidade Curricular do Bloco Livre A / Unrestricted Elective A	QAC	Semestral / Semester	168	depende da UC escolhida/dependent of choice	6	Optativa / Elective

(5 Items)

Mapa III - - 4º Ano / 8º Semestre**2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (M)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):**

<sem resposta>

2.3. Branches, options, profiles, major/minor, or other forms (if applicable):

<no answer>

2.4. Ano/semestre/trimestre curricular:

4º Ano / 8º Semestre

2.4. Curricular year/semester/trimester:

4th Year / 8th Semester

2.5. Plano de Estudos / Study plan

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Aspetos Socio-Profissionais da Informática / Social-Professional Aspects of Informatics	CHS	Semestral / Semester	84	TP:28	3	Obrigatória / Mandatory
Empreendedorismo / Entrepreneurship	CC	Semestral / Semester	80	TP:45	3	Obrigatória / Mandatory
Unidade de Consolidação VI / Consolidation Unit VI	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Unidade de Consolidação VII / Consolidation Unit VII	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Unidade de Especialização I / Specialization Unit I	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Unidade de Especialização II / Specialization Unit II	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective

(6 Items)

Mapa III - - 3º Ano / 6º Semestre e 4º Ano / 7 - 8º Semestre - Opções - Unidade Consolidação**2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (M)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):***<sem resposta>***2.3. Branches, options, profiles, major/minor, or other forms (if applicable):***<no answer>***2.4. Ano/semestre/trimestre curricular:***3º Ano / 6º Semestre e 4º Ano / 7 - 8º Semestre - Opções - Unidade Consolidação***2.4. Curricular year/semester/trimester:***3th Year / 6th Semester and 4th Year / 7- 8th Semester - Electives - Consolidation Unit***2.5. Plano de Estudos / Study plan**

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Análise e Desenho de Algoritmos / Design and Analysis of Algorithms	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Aprendizagem Automática / Machine Learning	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Concorrência e Paralelismo / Concurrency and Parallelism	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective

Construção e Verificação de Software / Software Construction and Verification		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Engenharia de Software / Software Engineering		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Interação Pessoa-Máquina / Human-Computer Interaction		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Interpretação e Compilação / Interpretation and Compilation		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Representação de Conhecimento e Raciocínio / Knowledge Representation and Reasoning		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Segurança de Sistemas Computacionais / Computing Systems Security		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Sistemas de Bases de Dados / Database Systems		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Sistemas Distribuídos / Distributed Systems		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective

(11 Items)

Mapa III - - 5º Ano / 9º Semestre

2.1. Ciclo de Estudos:

Engenharia Informática

2.1. Study Cycle:

Computer Science and Engineering

2.2. Grau:

Mestre (M)

2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):

<sem resposta>

2.3. Branches, options, profiles, major/minor, or other forms (if applicable):

<no answer>

2.4. Ano/semestre/trimestre curricular:

5º Ano / 9º Semestre

2.4. Curricular year/semester/trimester:

5th Year / 9th Semester

2.5. Plano de Estudos / Study plan

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Unidade de Especialização III / Specialization Unit III		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Unidade de Especialização IV / Specialization Unit IV		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Unidade de Especialização V / Specialization Unit V		Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective

(3 Items)

Mapa III - - 4º Ano / 8º Semestre e 5º Ano / 9º Semestre - Opção - Unidade de Especialização**2.1. Ciclo de Estudos:***Engenharia Informática***2.1. Study Cycle:***Computer Science and Engineering***2.2. Grau:***Mestre (MI)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):***<sem resposta>***2.3. Branches, options, profiles, major/minor, or other forms (if applicable):***<no answer>***2.4. Ano/semestre/trimestre curricular:***4º Ano / 8º Semestre e 5º Ano / 9º Semestre - Opção - Unidade de Especialização***2.4. Curricular year/semester/trimester:***4th Year / 8th Semester and 5th Year / 9th Semester - Electives - Specialization Unit***2.5. Plano de Estudos / Study plan**

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Unidade de Consolidação / Consolidation Unit	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Algoritmos e Sistemas Distribuídos / Distributed Algorithms and Systems	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Arquitetura e Implementação de Sistemas de Operação / Operating Systems Design and Implementation	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Codificação da Informação / Information Coding	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Programação com Restrições / Constraint Programming	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Computação de Alto Desempenho / High Performance Computing	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Computação Multimédia / Multimedia Computation	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Conceitos e Tecnologias XML / Concepts and Technologies of XML	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Data Warehousing / Data Warehousing	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Gestão de Centros de Dados / Data Center Management	I	Semestral / Semester	168	TP: 28	6	Optativa / Elective

Gestão de Projetos Informáticos / IT Project Management	I	Semestral / Semester	168	TP: 28	6	Optativa / Elective
Linguagens para Domínios Específicos / Domain Specific Languages	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Lógicas para a Programação e Especificação / Programming and Specification Logics	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Modelos de Concorrência e Segurança / Models of Concurrency and Security	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Pesquisa e Otimização / Search and Optimization	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Programação Multiparadigma / Multiparadigm Programming	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Qualidade do Software / Software Quality	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Redes de Computadores TCP/IP / TCP/IP Networking	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Requisitos e Arquitetura de Software / Software Requirements and Architecture	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Reutilização e Evolução de Software / Software Reuse and Evolution	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Segurança de Sistemas e Redes de Computadores / Computer Systems and Network Security	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Síntese de Imagem / Image Synthesis	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Sistemas de Computação em Cloud / Cloud Computing Systems	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Sistemas de Computação Móvel e Ubíqua / Mobile and Ubiquitous Computing	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Sistemas Multi-Agente / Multi Agent Systems	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Técnicas Avançadas de Análise e Desenho de Algoritmos / Advanced Algorithm Design and Analysis Techniques	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Tecnologias de Informação Geográfica / Geographic Information Technology	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective
Web Semântica / Semantic Web	I	Semestral / Semester	168	T:28 P: 28	6	Optativa / Elective

(28 Items)**Mapa III - - 5º Ano / 10º Semestre****2.1. Ciclo de Estudos:*****Engenharia Informática***

2.1. Study Cycle:*Computer Science and Engineering***2.2. Grau:***Mestre (MI)***2.3. Ramos, opções, perfis, maior/menor, ou outras (se aplicável):***<sem resposta>***2.3. Branches, options, profiles, major/minor, or other forms (if applicable):***<no answer>***2.4. Ano/semestre/trimestre curricular:***5º Ano / 10º Semestre***2.4. Curricular year/semester/trimester:***5th Year / 10th Semester***2.5. Plano de Estudos / Study plan**

Unidades Curriculares / Curricular Units	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Dissertação / Dissertation I (1 Item)		Anual / Annual	1176	OT:40	42	Obrigatória / Mandatory

3. Descrição e fundamentação dos objectivos**3.1. Dos objectivos do ciclo de estudos****3.1.1. Objectivos gerais do ciclo de estudos:**

O objetivo do Mestrado Integrado em Engenharia Informática é formar Engenheiros Informáticos de conceção habilitados a desenvolver atividades de projeto, liderança, e inovação, por vezes em contexto de investigação, e com bases para aceder a um 3º Ciclo. Como disposto no DL74/2006 para o grau de mestre, os graduados pelo curso deverão:

- Conhecer princípios e técnicas, e desenvolver capacidades analíticas que suportem a conceção e desenvolvimento, assim como inovação tecnológica e científica.*
- Saber aplicar os conhecimentos na resolução de novos problemas, incluindo em contextos multidisciplinares.*
- Saber pesquisar e integrar conhecimentos, avaliar e comparar soluções criticamente.*
- Revelar capacidade para refletir sobre implicações éticas e sociais da atividade, numa perspetiva de Ciência, Tecnologia e Sociedade, assim como para comunicar resultados de forma clara e rigorosa;*
- Demonstrar competências e atitudes colaborativas, que suportem a aprendizagem ao longo da vida.*

3.1.1. Study cycle's generic objectives:

The objective of the integrated Master program in CSE is to educate conception-level Computer Science Engineers prepared to develop project, leadership, and innovation activities, many times in a R&D context, and with basis to pursue a 3th cycle. As stipulated in DL74/2006 for the master degree, program graduates should be:

- Knowledgeable of principles, techniques and analytical skills supporting design and conceptual engineering activities, as well as technological and scientific innovation.*
- Able to apply knowledge in the solution of new problems, including in multidisciplinary contexts.*
- Able to research and integrate knowledge, and critically evaluate and compare solutions.*
- Demonstrate ability to reflect on the ethical and social consequences of their activity, in a perspective of Science, Technology and Society, and to communicate results with clarity and technical rigour.*
- Develop collaborative attitude and team work competencies, able to support life long learning skills.*

3.1.2. Objectivos de aprendizagem (conhecimentos, aptidões e competências) a desenvolver pelos estudantes:

Compreender os fundamentos e o estado-da-arte em Engenharia Informática de conceção, utilizar bases matemáticas e técnicas, e métodos científicos e de engenharia no desenvolvimento de atividades profissionais; Lidar com as várias facetas dos sistemas informáticos, aliando capacidades orientadas para o desenvolvimento técnico, às de interpretação do contexto social em que estes se inserem, num espírito de autoatualização contínua;

Enfrentar a complexidade de forma proactiva, demonstrando criatividade.

Trabalhar em equipa, saber documentar objetivos, soluções e resultados a especialistas e a não especialistas;

Endereçar aspetos ético-profissionais, incluindo responsabilidades sociais e ambientais;

Relatar, sintetizar, e argumentar temas técnicos, demonstrar visão crítica perante soluções propostas;

Desempenhar atividade profissional em Engenharia Informática, a partir de bases de preparação técnica e científica que possam alicerçar uma orientação para o empreendedorismo.

3.1.2. Intended learning outcomes (knowledge, skills and competences) to be developed by the students:

Understand the foundations and state-of-the art in Computer Science and Engineering, using adequate mathematical and technical instruments, as well as engineering and scientific methods in the professional activities;

Dominate the several facets of informatic systems, adding to technical development skills the ability to interpret the social context in which such systems are embedded, promoting a knowledge improvement, lifelong learning attitude;

Deal with the complexity with pro activity, demonstrating creativity;

Work in a team, know how to document objectives, solutions and results, both to specialists and non-specialists;

Address ethical-professional aspects, including social / environmental responsibilities;

Relate, synthesise, and argue technical themes, develop a critical view before proposed solutions;

Develop a professional activity in CSE, building on a strong technical and scientific basis that may also be leveraged on entrepreneurship oriented activities.

3.1.3. Coerência dos objectivos definidos com a missão e a estratégia da Instituição de Ensino:

A FCT/UNL é uma instituição de ensino superior universitário dirigida às áreas de Ciência e de Engenharia, que tem como missão desenvolver:

a) Investigação científica competitiva no plano internacional, privilegiando áreas interdisciplinares, incluindo a investigação orientada para a resolução de problemas que afetam a sociedade;

b) Um ensino de excelência, com ênfase crescente na investigação realizada, veiculado por programas académicos competitivos a nível nacional e internacional;

c) Uma base alargada de participação interinstitucional orientada para a integração das diferentes culturas científicas, com vista à criação de sinergias inovadoras para o ensino e para a investigação;

d) Uma forte ligação à sociedade, transferência de conhecimentos, tecnologias e serviços, quer no plano interno, quer no plano internacional, capaz de contribuir para o desenvolvimento social e para a qualificação dos recursos humanos.

O curso de Mestrado Integrado em Engenharia Informática contribui para a formação de Engenheiros com o nível adequado ao desenvolvimento de projeto e de atividades de inovação. Acresce que dado o carácter de “enabling technology” cada vez mais adquirido pela Informática e o seu crescente impacto, estamos em presença de um setor em expansão, onde continua a existir uma enorme procura de recursos humanos cada vez mais sofisticados, necessários para a criação de produtos e serviços inovadores e competitivos, geradores de vantagens económicas em todas as áreas da sociedade.

É assim uma aposta estratégica da escola continuar a contribuir decisivamente para o desenvolvimento da educação avançada e investigação em Engenharia Informática, como faz há 37 anos, como escola pioneira na área no nosso País.

É de referir o alinhamento do curso com o chamado Perfil Curricular da FCT, que favorece o desenvolvimento de várias competências transversais, na área da comunicação, ciência tecnologia e sociedade, e empreendedorismo, potenciando a ligação da escola à sociedade em geral através da oferta de atividades colaborativas com o exterior, em particular com as empresas, por exemplo através da oferta de estágios-internships, e dissertações em regime académico-empresarial.

Finalmente, o curso contribui ainda para formar estudantes preparados para aceder um 3º ciclo (Doutoramento) em Ciência e Engenharia Informática, potenciando a investigação realizada no Departamento de Informática e nos seus Centros de Investigação. Para este objetivo contribui a possibilidade dos estudantes desenvolverem a sua dissertação em ambiente de investigação científica (mais de 250 dissertações deste 2007), e de se envolverem em atividades de investigação ao longo do curso, através do Programa de Iniciação à Investigação, da unidade de Atividade Prática de Desenvolvimento Curricular, e da preocupação constante em dar visibilidade à fronteira do conhecimento em inúmeras unidades curriculares.

3.1.3. Coherence of the defined objectives with the Institution's mission and strategy:

FCT/UNL is a higher education university institution in the Science and Engineering area. Its mission comprises developing:

- a) Internationally competitive scientific research, with an emphasis in interdisciplinary, including research oriented towards the solution of societal challenges.*
- b) Excellence in teaching, supported on research activities; implemented by academic programs, competitive at the national and international level;*
- c) A broad basis for interinstitutional participation, towards the integration of different scientific cultures, fostering the creation of synergies for research and education.*
- d) A strong social commitment, promoting technology, service and knowledge transfer, able to contribute for the social development and for the high quality qualification of human resources.*

The integrated Master program now proposed contributes for the education of engineers in CSE, with a level of competence adequate for project development and innovation activities. Moreover, given the enabling technology character of Informatics in general, and its increasing impact in society, we are in presence of a continuously expanding sector, where the demand for increasingly sophisticated human resources is steadily growing. Such human resources are necessary to support the creation of innovative and competitive products and services, able to generate economical growth in all areas of society.

Therefore, FCT UNL is strategically strongly committed to continue contributing for the development of advanced education and research in Computer Science and Engineering (Informática), as has been doing for 37 years now, as a pioneering school in the field at the national level.

The MSc program is perfectly aligned with the so-called "FCT Curricular Profile", which favours the emergence of various transferable skills in areas such as communication, science, technology and society, and entrepreneurship, leveraging the connections between the school and the society at large, in particular, with firms and companies, for instance by offering to students internships and master dissertations in academic-industrial collaboration.

Finally, the program also contributes for the education of strong candidates to a 3th cycle (PhD program) in Computer Science, leveraging on the research pursued in the CS Department and its research units. This goal is enforced by the opportunity offered to students of developing their master dissertation in the context of a departmental research activity (we have delivered more than 250 master thesis since 2007), and of getting involved in research during the program, through the UROP (Undergraduate Research Opportunities program), the unit "Practical Activity for Curricular Development", and also due to our always present concern of providing effective visibility to the frontier of knowledge in so many curricular units.

3.2. Adequação ao Projecto Educativo, Científico e Cultural da Instituição

3.2.1. Projecto educativo, científico e cultural da Instituição:

De acordo com os seus Estatutos, a Faculdade de Ciências e Tecnologia tem identidade e missão idênticas às da UNL, dirigidas às áreas de Ciências e de Engenharia. Na sua missão, enquanto instituição universitária que se pretende de referência, inclui-se o desenvolvimento de investigação competitiva no plano internacional, privilegiando áreas interdisciplinares e a investigação orientada para a resolução dos problemas que afetam a sociedade, bem como a oferta de ensino de excelência, com ênfase crescente em segundos e terceiros ciclos, mas fundado em primeiros ciclos sólidos, veiculado por programas académicos competitivos a nível nacional e internacional, erigindo o mérito como medida essencial da avaliação.

Fundamentalmente, a política de ensino e investigação tem por objetivo promover a qualidade e reconhecimento destas atividades, devendo a investigação ser progressivamente incorporada nas estruturas curriculares dos ciclos de estudos, proporcionando uma oferta educativa atualizada e substancialmente diferenciadora. Por outro lado, a Faculdade dispõe de uma política de qualidade que visa assegurar a melhoria contínua das suas atividades, por forma a aumentar, de modo sustentado, a sua eficiência e corresponder às expectativas decorrentes do seu objeto social.

Neste âmbito, o projeto educativo tem contemplado não só a criação de novas áreas de estudo, decorrentes da evolução da economia associada às mudanças sociais, como também a introdução de métodos de ensino e de avaliação conducentes a uma aprendizagem mais eficiente e a reestruturação da oferta formativa existente. Neste último caso, salienta-se a recente introdução, em todos os cursos de Licenciatura, de Mestrado e de Mestrado Integrado, de competências complementares, designadamente soft skills, contacto com empresas ou investigação e empreendedorismo, configurando o designado "Perfil Curricular FCT" como fator diferenciador dos diplomados da instituição e elemento facilitador da sua inserção na vida ativa (<http://www.fct.unl.pt/perfil-curricular-fct>).

Com o intuito de progredir para uma escola "research oriented", a Faculdade tem vindo a adotar uma política de incentivos para o desenvolvimento de atividades de investigação, potenciando o mérito dos seus docentes como referencial e, ainda, uma política promotora de transferência da tecnologia e do conhecimento gerados para a Sociedade através de parcerias com empresas, licenciamento de propriedade industrial e apoio à criação de empresas spin-off.

A Faculdade atribui grande importância às atividades culturais que disponibiliza aos seus estudantes,

considerando que valorizam a qualidade dos serviços educativos que oferece e que constituem elemento diferenciador para a notoriedade da Escola. Assim, para cada ano letivo é programado um extenso conjunto de atividades culturais de alto nível (palestras, conferências, debates, exposições de arte) com a intervenção de personalidades detentoras de elevado prestígio nacional e internacional.

3.2.1. Institution's educational, scientific and cultural project:

According to its Statutes, the Faculty of Sciences and Technology identity and mission in the Sciences and Engineering areas are similar to those of the UNL – Universidade NOVA de Lisboa. As a higher education institution striving to be a reference, it includes the development of competitive research at international level that privileges interdisciplinary areas and research aimed at solving social problems, as well as an educational excellence offer increasingly focused on second and third cycles, but founded on solid first cycles with competitive academic programs at both national and international levels, adopting merit as the essential measure of assessment.

Basically the policy for teaching and research aims at promoting quality and recognition of those activities, increasingly incorporating research in the curricular structures of the study cycles, enabling an updated educational offer expected to be positively discriminated. On the other hand, the Faculty is enforcing a quality policy for the continuous improvement of its activities in order to increment its efficiency in a sustainable process leading to a better achievement of its social responsibilities. Therefore, its educational project includes not only the creation of new study areas that can follow economical evolution associated to social changes but also the introduction of teaching and assessment methods aimed at improving the learning efficiency, and the restructuring of the existent educational offer. About this last issue, it is worth mentioning the recent introduction, in all first and second study cycles and Integrated Master programs, of common competences, namely soft skills, undergraduate practice or research opportunities and entrepreneurship, leading to the so-called “Perfil Curricular FCT” (FCT Curricular Profile) as a differentiating feature of the institution graduates and a facilitator of their insertion in the active life (<http://www.fct.unl.pt/perfil-curricular-fct>).

As the Faculty aims to become a research oriented school, a policy of incentives to research development is being adopted fostering the merit of its academic staff and, also, a policy aims at promoting the technology and knowledge transfer to the Society through partnerships with companies, licensing of industrial property and support to the creation of spin-off companies.

Cultural activities are looked as an important aspect of the Faculty's educational offer that contributes to a positive discrimination of the School. For each academic year a set of high-level cultural activities is scheduled, such as seminars, conferences, debates and art exhibitions, with the cooperation of prominent individualities holding high national and international prestige.

3.2.2. Demonstração de que os objectivos definidos para o ciclo de estudos são compatíveis com o projecto educativo, científico e cultural da Instituição:

A Faculdade de Ciências e Tecnologia da UNL inclui na sua missão o desenvolvimento de investigação competitiva no plano internacional, bem como a oferta de ensino de excelência, com ênfase crescente em segundos e terceiros ciclos, fundado em primeiros ciclos sólidos, veiculado por programas académicos competitivos a nível nacional e internacional.

A oferta educativa da FCTUNL encontra-se, a nível de licenciaturas e mestrados nas Engenharias, essencialmente organizada sob a forma de Mestrados Integrados, respondendo aos requisitos de formação das áreas respetivas. Os objetivos do Mestrado Integrado em Engenharia Informática aqui proposto, integram de forma coesa e flexível os objetivos de um 1º e 2º ciclo na área, e visam a formação de Engenheiros Informáticos habilitados a desenvolver atividades profissionais de projeto, liderança, e inovação, em muitos casos em contexto de investigação, com bases para aceder a um 3º Ciclo, de acordo o Artº 15 do DL74/2006. Esta orientação é implementada numa lógica de continuidade, sendo que os eventuais diplomados de 1º ciclo adquirem uma formação sólida, coerente e relevante em termos de mobilidade.

Tais objetivos estão claramente alinhados com o projeto educativo, científico e cultural da FCT UNL.

O curso adequa-se ao “Perfil Curricular da FCT”, conjunto de características comuns a todos os cursos da escola que favorecem o desenvolvimento de competências transversais, potenciam a ligação à sociedade, e desenvolvem uma cultura de inovação, empreendedorismo, e desenvolvimento científico. Em harmonia com a investigação realizada no Departamento de Informática e nos seus Centros, o curso contribui ainda para formar estudantes bem preparados para aceder a cursos de 3º ciclo em Informática, em particular no seu programa doutoral (atualmente com cerca de 75 alunos).

São inúmeras as áreas utilizadoras de engenheiros de conceção, com formação integral de dois ciclos em Engenharia Informática: desenvolvimento de software; telecomunicações, media, internet; produtores de equipamentos; operadores de sistemas de informação; consultoria; inovação, investigação e desenvolvimento; e atividades de desenvolvimento de serviços e produtos em áreas tão diversas como a energia, a gestão empresarial, a medicina, o ambiente, os transportes, os conteúdos, o entretenimento, e tantas outras.

É uma aposta estratégica da FCTUNL continuar a contribuir decisivamente para o desenvolvimento da educação avançada e investigação científica e inovação em Informática, como escola pioneira na área no nosso País, tendo formado mais de 1400 licenciados pré-Bolonha e mestres de 2º ciclo nos últimos 10 anos.

3.2.2. Demonstration that the study cycle's objectives are compatible with the Institution's educational, scientific and cultural project:

The Faculdade de Ciências e Tecnologia da UNL (FCT UNL) includes in its mission the development of internationally competitive research, excellence in teaching with an emphasis on 2nd and 3th cycles, while being founded in solid 1st cycles, implemented in nationally and internationally competitive academic programs. FCTUNL educational offer is, at the level of the engineering programs and mostly structured under the form of integrated masters, answering to the requirements of the respective areas.

The objectives of the now proposed integrated master in computer science and engineering integrate, in a tightly coupled yet flexible way, the goals of a 1st and 2nd cycle in the area, towards the education of engineers able to develop professional activities involving project, leadership, innovation, often in a R&D context, and potential candidates for a PhD, according to the Artº15 of DL74/2006. These guidelines are implemented in a progressive scheme along the 5 years, so that possible 1st cycle graduates may receive a sufficiently solid, coherent and relevant education, useful for mobility purposes.

These objectives are clearly aligned with the educational, scientific and cultural project of FCT UNL.

The program conforms to the "FCT Curricular Profile", which favours the emergence of various transferable skills in areas such as communication, science, technology and society, and entrepreneurship, leveraging the connections between the school and the society at large, and promote an culture of innovation, entrepreneurship, and scientific development. In harmony with the Departmental research activities, the master program contributes to the education of potentially strong candidates to a 3th cycle, in particular to our local PhD program in CSE (currently with 75 students in the pipeline). The demand of conception-oriented engineers with an integral education corresponding to two cycles appear in just too many areas: software development, telecommunications, media, Internet, production of equipment, information systems construction and operation, consulting, innovation, R&D; and activities related to the development of services and products in areas so diverse as energy, management, medicine, environment, transport systems, entertainment, and so many others.

It is a strong strategical commitment of FCT UNL to continue contributing for the development of advanced education and research in Computer Science and Engineering (Informática), as has been doing for 37 years now, as a pioneering school in the field at the national level, having delivered more than 1400 pre bolonha 5-year "licenciatura" graduates and 2nd cycle masters in the last 10 years.

3.3. Unidades Curriculares

Mapa IV - Análise Matemática I E / Mathematical Analysis I E

3.3.1. Unidade curricular:

Análise Matemática I E / Mathematical Analysis I E

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Bento José Carrilho Miguens Louro (apenas responsável, não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Ana Luísa da Graça Batista Custódio (Regente) (T-84h; P-42h)

João Filipe Lita da Silva (P-126h)

Pamela Pacciani (P-126h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

No final desta unidade curricular o estudante terá adquirido conhecimentos, aptidões e competências que lhe permitam:

- Trabalhar com noções elementares de topologia na recta real (vizinhança, aberto, fechado, etc.);*
- Fazer pequenas demonstrações por indução;*
- Compreender a noção rigorosa de limite (de sucessões, de funções de variável real) e calcular limites.*
- Compreender a noção rigorosa de continuidade de funções de variável real e respectivos resultados fundamentais.*
- Conhecer a noção de diferenciabilidade, os teoremas de Rolle, Lagrange e Cauchy e aplicações ao cálculo de limites;*
- Conhecer o desenvolvimento de Taylor e aplicações ao estudo de funções;*
- Conhecer a noção de primitiva e respectivas técnicas de cálculo;*
- Conhecer a noção de integral de Riemann, respectivas técnicas de cálculo e algumas aplicações;*
- Ser capaz de estudar a convergência de integrais impróprios.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

At the end of this course the student must have acquired knowledge, skills and powers to:

- *Work with elementary notions of topology on the real line (neighborhood, open, closed, etc.).*
- *Make small proves by induction;*
- *Understand the concept and definition of limit (sequences, functions of real variable) and calculate limits.*
- *Understand the definition of continuity of functions of one real variable and the fundamental results.*
- *Understand the notion of differentiability, the theorems of Rolle, Lagrange and Cauchy and their applications to the calculation of limits;*
- *Understand the Taylor development and its applications to the study of functions;*
- *Understand the notion of indefinite integral and perform calculations;*
- *Understand the notion of Riemann integral, the techniques of calculation and some applications;*
- *Be able to study the convergence of improper integrals.*

3.3.5. Conteúdos programáticos:

1. *Topologia - Indução Matemática – Sucessões: Topologia elementar da recta real. Relação de ordem na recta real. Princípio de indução matemática. Generalidades sobre sucessões. Noção de convergência de uma sucessão e propriedades do cálculo de limites. Subsucessões. Teorema de Bolzano-Weierstrass.*
2. *Limites e Continuidade: Limite segundo Cauchy e Heine. Propriedades de cálculo. Continuidade de uma função num ponto. Propriedades das funções contínuas. Teorema do valor intermédio. Teorema de Weierstrass. Continuidade e bijecções recíprocas.*
3. *Diferenciabilidade: Generalidades. Teoremas fundamentais: Rolle, Lagrange e Cauchy. Cálculo prático de limites. Desenvolvimento de Taylor e aplicações.*
4. *Primitivação: Introdução. Primitivação por partes. Primitivação por substituição. Primitivação de funções racionais.*
5. *Integração de Riemann: Introdução. Teoremas fundamentais. Integração por partes e integração por substituição. Aplicações diversas. Integrais impróprios.*

3.3.5. Syllabus:

1. *Topology - Mathematical Induction – Sequences: Basic topology of the real numbers. Order relation. Mathematical induction. Generalities about sequences. Convergence of a sequence and properties for calculus of limits. Subsequences. Bolzano-Weierstrass theorem.*
2. *Limits and Continuity: Convergence according to Cauchy and Heine. Calculus properties. Continuity of a function at a given point. Properties of continuous functions. Bolzano theorem. Weierstrass theorem. Continuity and reciprocal bijections.*
3. *Differentiability: Generalities. Fundamental theorems: Rolle, Lagrange and Cauchy. Calculus techniques for limits. Taylor formula and applications.*
4. *Indefinite Integration: Introduction. Indefinite integration by parts. Indefinite integration by substitution. Indefinite integration of rational functions.*
5. *Riemann Integration: Introduction. Fundamental theorems. Definite integration by parts and by substitution. Some applications. Improper integration.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O capítulo 1 é dedicado às noções topológicas, indução matemática e sucessões de números reais. Cobrem-se, assim os dois primeiros objetivos enunciados e parte do terceiro.

O capítulo 2 é dedicado aos limites e continuidade de funções reais de variável real, cobrindo parte do terceiro objetivo e o quarto.

O capítulo 3 é dedicado ao estudo do cálculo diferencial de funções reais de variável real e resultados fundamentais, cobrindo o quinto objetivo. O estudo da fórmula de Taylor e aplicações cobre o sexto objetivo.

O capítulo 4 é dedicado ao estudo das primitivas e respectivas técnicas de cálculo, cobrindo o sétimo objetivo.

O capítulo 5 é dedicado ao estudo do integral de Riemann e dos integrais impróprios, cobrindo os oitavo e nono objetivos.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Chapter 1 is devoted to topological notions, mathematical induction and sequences of real numbers. It covers the first two objectives and part of the third.

Chapter 2 is devoted to the study of limits and continuity of real functions of one real variable, covering part of the third and the fourth objective.

Chapter 3 is devoted to the study of differential calculus of real functions of one real variable and main results, covering the fifth objective. The study of Taylor formula and its applications covers the sixth objective.

Chapter 4 is devoted to the study of indefinite integrals and their calculation techniques, covering the seventh objective.

Chapter 5 is devoted to the study of the Riemann integral and improper integrals, covering the eighth and ninth objectives.

3.3.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas consistem na exposição da matéria, que é ilustrada com exemplos de aplicação.

As aulas práticas consistem na resolução de exercícios de aplicação dos métodos e resultados apresentados nas aulas teóricas.

Quaisquer dúvidas são esclarecidas no decorrer das aulas, nas sessões semanais destinadas ao atendimento dos estudantes ou ainda em sessões combinadas diretamente entre aluno e professor.

O estudante deve assistir a todas aulas práticas, com possível exceção de três.

O estudante pode realizar a disciplina por avaliação contínua que consiste na realização de três testes e na avaliação pelo docente das aulas práticas. Em caso de insucesso, o estudante pode ainda apresentar-se a exame.

3.3.7. Teaching methodologies (including evaluation):

Theoretical classes consist in a theoretical exposition illustrated by application examples.

Practical classes consist in the resolution of application exercises for the methods and results presented in the theoretical classes.

Students can ask questions during the classes, in weekly scheduled sessions or in special sessions accorded directly with the professor.

Students must attend classes, with the possible exception of three.

There are three mid-term tests that can substitute the final exam in case of approval. Otherwise the student must pass the final exam.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas teóricas procede-se à exposição da matéria, ilustrada com exemplos. Em geral, os resultados são explicados e exemplificados, sem demonstração formal. No entanto, são feitas algumas demonstrações, especialmente quando estas são úteis para a melhor compreensão da matéria.

Os alunos têm acesso a uma lista de problemas resolvidos, de problemas para resolver nas aulas práticas e ainda de problemas para resolver autonomamente.

Para obter aprovação, o aluno deve assistir a, pelo menos, dois terços das aulas práticas. Esta prática tem-se revelado útil, especialmente para os alunos de primeira inscrição na Universidade, impedindo a abstenção às aulas e respectivas consequências.

Além trabalhar os conceitos expostos na teórica, os problemas propostos também têm como objectivo importante a prática do cálculo.

É atribuída uma classificação nas aulas práticas, que é tida em conta na classificação final.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In theoretical classes matters are explained and illustrated with examples. In general, results are explained and exemplified, without a formal proof. Nevertheless, some proofs are given, especially when they are useful to understand the matter.

Students can obtain a list of solved problems, problems to be solved in practical classes and problems to solve by themselves.

In order to succeed the student must attend, at least, two thirds of the classes. Such practice has revealed to be useful, mainly to the first year students.

Besides working on the concepts set out in theoretical classes, proposed problems have also as an important objective, the practice of calculation.

It is given a rating in practical classes, which is taken into account in the final standings.

3.3.9. Bibliografia principal:

Texto Adoptado

Ana Alves de Sá e Bento Louro, Análise Matemática I, FCT-UNL, 2011

Bibliografia Recomendada

- 1. Robert G. Bartle e Donald R. Sherbert, Introduction to Real Analysis, John Wiley & Sons Inc., 1999*
- 2. Jaime Campos Ferreira, Introdução à Análise Matemática, Fundação Calouste Gulbenkian, 1982*
- 3. Rod Haggarty, Fundamentals of Mathematical Analysis, Prentice Hall, 1993*
- 4. Carlos Sarrico, Análise Matemática, Leituras e Exercícios, Gradiva, 1997*

Mapa IV - Álgebra Linear e Geometria Analítica E / Linear Algebra and Analythic Geometry E

3.3.1. Unidade curricular:

Álgebra Linear e Geometria Analítica E / Linear Algebra and Analythic Geometry E

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Maria de Fátima Vale de Gato Santos Rodrigues (Responsável e Regente) (T - 42h; P - 56h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Isabel Maria Oitavem Fonseca da Rocha Kahle (P- 28h)

Bruno Rodrigues Vaz (P-56h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

No final desta unidade curricular é esperado que os estudantes consigam:

- Operar com matrizes, calcular a inversa de uma matriz.*
- Resolver e analisar um sistema de equações aplicando o conhecimento matricial discutindo a sua possibilidade de solução.*
- Calcular o valor do determinante de uma matriz.*
- Identificar vectores linearmente independentes, geradores e bases de um dado espaço vectorial.*
- Determinar a intersecção e a soma de dois subespaços.*
- Reconhecer aplicações lineares identificando o seu núcleo e imagem.*
- Construir a matriz de uma aplicação linear e matrizes de mudança de base.*
- Calcular valores e vectores próprios de matrizes e de endomorfismos.*
- Identificar matrizes diagonalizáveis.*
- Calcular o produto interno externo e misto de vectores.*
- Calcular distâncias e ângulos envolvendo pontos, rectas e planos.*
- Determinar a posição relativa entre pontos, rectas e planos recorrendo ao conhecimento de análise sobre sistemas de equações.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

At the end of this course is expected that students be able to:

- Operate with matrices; compute the inverse of a matrix.*
- Solve and analyze a system of equations by applying the matrix knowledge discussing the possibility of their solution.*
- Calculate the value of the determinant of a matrix.*
- Identify vectors linearly independent, generators and basis of a given vector space.*
- Determine the intersection and the sum of two subspaces.*
- Recognize linear maps and identify its kernel and its range.*
- Building a linear map matrix and a change of basis matrix.*
- Compute eigenvalues and eigenvectors of a matrix and of an endomorphism.*
- Identify diagonalizable matrices.*
- Calculate the dot and the cross product of vectors.*
- Calculate distances and angles involving points, lines and planes.*
- Determine the relative position between points, lines and planes using the knowledge of analysis of systems of equations.*

3.3.5. Conteúdos programáticos:

Matrizes, Sistemas de Equações Lineares, Determinantes, Espaços Vectoriais, Aplicações Lineares, Valores e Vectores Próprios, Produto Interno, Externo e Misto, Geometria Analítica.

3.3.5. Syllabus:

Matrices, Systems of linear equations, Determinants, Vectorial Spaces, Linear transformations, Eigenvectors and eigenvalues, Inner product and vector product. Analytic geometry.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O programa consta de conceitos básicos de Álgebra Linear e Geometria Analítica. São estudados os teoremas mais importantes do cálculo matricial, espaços vectoriais e aplicações lineares bem como vectores e valores próprios.

São ainda explorados os conceitos de produto interno e externo para vectores de R^3 .

No que diz respeito à Geometria Analítica são estudadas as representações analíticas da recta e do plano, as suas posições relativas e são explorados os problemas métricos de distâncias e ângulos, em R^2 e em R^3 .

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The program consists on basic concepts of Linear Algebra and Analytic Geometry. The main theorems of matrix theory, vector spaces, linear transformations and eigenvalues and eigenvectors are studied, including applications.

Furthermore the dot product and cross product concepts in R^3 are explored.

Analytic representation of the straight line and the plane, their relative positions and some problems of distances and angles are studied, in R^2 and R^3 .

3.3.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas consistem na exposição da matéria, que é ilustrada com exemplos de aplicação.

As aulas práticas consistem na resolução de exercícios de aplicação dos métodos e resultados apresentados nas aulas teóricas.

Quaisquer dúvidas são esclarecidas no decorrer das aulas, nas sessões semanais destinadas ao atendimento dos alunos ou ainda em sessões combinadas directamente entre aluno e professor.

Realizam-se quatro testes durante o semestre, que dispensam de exame em caso de média positiva. Se não dispensou, o aluno deve apresentar-se a exame.

3.3.7. Teaching methodologies (including evaluation):

Theoretical classes consist in a theoretical exposition illustrated by application examples.

Practical classes consist in the resolution of application exercises for the methods and results presented in the theoretical classes.

Students can ask questions during the classes, in weekly scheduled sessions or in special sessions accorded directly with the professor.

There are four mid-term tests that can substitute the final exam in case of approval. Otherwise the student must obtain approval in the final exam.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas teóricas procede-se à exposição da matéria, ilustrada com exemplos. Em geral, os resultados são explicados e exemplificados, sem demonstração formal. No entanto, são feitas algumas demonstrações, especialmente quando estas são úteis para a melhor compreensão da matéria.

Para obter aprovação, o aluno não pode faltar a mais de duas das aulas práticas leccionadas. Esta prática tem-se revelado útil, especialmente para os alunos de primeira inscrição na Universidade, impedindo a abstenção às aulas e respectivas consequências.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In theoretical classes matters are explained and illustrated with examples. In general, results are explained and exemplified, without a formal proof. Nevertheless, some proofs are given, especially when they are useful to understanding the matter.

In order to succeed the student must attend the classes (only two absences allowed). Such practice has revealed to be useful, mainly to the first year students.

3.3.9. Bibliografia principal:

ISABEL CABRAL, CECÍLIA PERDIGÃO, CARLOS SAIAGO, Álgebra Linear, Escolar Editora, 2012 (3ª Edição Revista e Actualizada)

T. S. BLYTH e E. F. ROBERTSON, Essential student algebra. Volume two: Matrices and Vector Spaces, Chapman and Hall, 1986.

T. S. BLYTH e E. F. ROBERTSON, Basic Linear Algebra (Springer undergraduate mathematics series), Springer, 1998. S. J. LEON, Linear Algebra with Applications, 6th Edition, Prentice Hall, 2002.

J. V. DE CARVALHO, Apontamentos da disciplina de Álgebra Linear e Geometria Analítica, Departamento de Matemática, Universidade Nova de Lisboa (ano lectivo 2000/2001)

Mapa IV - Introdução à Programação / Introductory Programming

3.3.1. Unidade curricular:

Introdução à Programação / Introductory Programming

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luis Manuel da Costa Caires (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Miguel Carlos Pacheco Afonso Goulão (Regente) (TP-126h)

Ana Maria Dinis Moreira (TP-63h)

António Maria Lobo César Alarcão Ravara (TP-63h; P-63h)

João Manuel dos Santos Lourenço (P-21h)

Fernanda Maria Barquinha Tavares Vieira Barbosa (TP-63h; P-63h)

Miguel Jorge Tavares Pessoa Monteiro (TP-126h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber: O significado das construções de programação incluídas no fragmento coberto da linguagem Java. Construir uma pequena aplicação no fragmento coberto a partir de uma especificação informal, mas rigorosa, escrita em Português, e usando a metodologia definida na disciplina. Quais os componentes e ferramentas básicas de um ambiente de desenvolvimento de software (editor, compilador, etc.) e qual a sua função.

Saber fazer: Desenvolver programas de pequena dimensão, bem organizados e obedecendo a um certo número de convenções. Conseguir projetar e escrever corretamente algoritmos simples. Ler e explicar / simular mentalmente a funcionalidade de fragmentos de código escritos no fragmento coberto da linguagem Java. Utilizar corretamente, ao nível esperado, as ferramentas de programação, assim como interpretar os seus resultados (mensagens de erro, etc.). Realizar, em grupo, um mini-projeto de desenvolvimento de software, integrando as várias competências transmitidas.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge: The meaning of the programming constructs included in the Java language fragment covered in the course. How to build a small application, using the covered Java language fragment, and using the methodology defined in this course. Know the components and basic tools of a software development environment (editor, compiler, etc) and their role.

Application: Develop well-organized, small-sized, programs, following a given set of standards. Project and write correctly simple algorithms. Read and explain / mentally simulate the functionality of code fragments written in the Java programming language. Correctly use, to the expected level, programming tools, as well as interpret their results (error messages, etc). Develop as a team, a software development mini-project, using the skills acquired in this course.

3.3.5. Conteúdos programáticos:

1. Computadores e programas

2. *Objetos e operações*
3. *Classes e tipos de dados básicos*
4. *Manipulação de estado*
5. *Mecanismos de decisão*
6. *Estrutura de aplicações*
7. *Interação (I/O)*
8. *Mecanismos de iteração*
9. *Manipulação de ficheiros*
10. *Aplicações com várias classes*
11. *Vetores e algoritmos relacionados*
12. *Ordenação e pesquisa*
13. *Metodologias de desenvolvimento de software*

3.3.5. Syllabus:

1. *Computers and programs*
2. *Objects and operations*
3. *Classes and basic data types*
4. *State manipulation*
5. *Decision constructs*
6. *Structure of an application*
7. *User interaction (I/O)*
8. *Iteration constructs*
9. *File manipulation*
10. *Applications with several classes*
11. *Vectors and related algorithms*
12. *Sorting and searching*
13. *Basic software development principles*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

As construções de programação do fragmento coberto da linguagem Java são apresentadas nos pontos 2 a 11 do conteúdo programático, sendo que a exposição teórica é complementada com a resolução de muitos exercícios.

Os alunos têm contacto com um ambiente de desenvolvimento de software desde a primeira aula, aprendendo incrementalmente a utilizar as ferramentas básicas do ambiente de desenvolvimento (editor, compilador, etc.) através da resolução de exercícios de programação.

A partir do ponto 6 do conteúdo programático os alunos aprendem a desenvolver aplicações muito simples e são abordadas metodologias básicas de desenvolvimento de software.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The programming constructs of the Java language fragment covered in the course are presented through syllabus topics 2 to 11, and are implementation of many exercises. Students are introduced to a software development environment on their first lecture, and incrementally learn how to use the basic tools of the development environment (editor, compiler, etc.) by solving programming problems. From syllabus topic 6 on, students learn how to develop simple applications and basic software development principles are covered.

3.3.7. Metodologias de ensino (avaliação incluída):

O ensino consiste em aulas teórico-práticas e aulas práticas (ambas em laboratório). Nas aulas teórico-práticas a exposição de matéria é intercalada com a resolução de exercícios nos quais os alunos aplicam os conceitos apresentados.

Nas aulas práticas, os alunos resolvem exercícios de consolidação dos conceitos apreendidos e praticados durante as aulas teórico-práticas.

A avaliação é composta por três componentes: dois testes (T1 e T2) ou um exame (NT), um trabalho prático (NP), e a nota de assiduidade (NA). A nota de assiduidade é o resultado da avaliação do trabalho do aluno durante as aulas e poderá incluir: assiduidade, participação durante as aulas e entrega de trabalhos de casa. Condições para obter aprovação: a média pesada das 3 componentes de avaliação for superior a 9.5; NP >= 9.5; e NT >= 9.5.

3.3.7. Teaching methodologies (including evaluation):

Lectures include theoretical exposition, which is followed by the resolution of problems that apply the concepts

discussed.

In the laboratory sessions, students solve consolidation problems of the concepts learned in the lectures.

Assessment has three components: two tests or a final exam (NT), practical assignment (TP), and a participation mark (NA). The participation mark is granted through the evaluation of the work done during the classes. It may include the assiduity, class participation, and homework.

Conditions for approval: weighted mean of all assessment components is greater or equal to 9.5; NP \geq 9.5; e NT \geq 9.5.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As aulas teórico-práticas são lecionadas em turnos de pequena dimensão (no máximo 28 alunos). Este topologia de aulas tem várias vantagens. Primeiro, nestas aulas os alunos aplicam imediatamente os conceitos apresentados durante a exposição teórica. Desta forma, qualquer duvida/dificuldade do aluno sobre a matéria exposta pode ser endereçada pelo docente prontamente. Segundo, permite o ensino coordenado de alunos com diferentes níveis de conhecimento e ritmos de aprendizagem. Isto porque, embora a exposição teórica tenha que ser sincronizada, a resolução de exercícios pode ser adaptada aos diferentes níveis de conhecimento dos alunos. Em geral, para cada conceito apresentado existe um exercício core e uma coleção de exercícios de diferentes graus de dificuldade: o exercício core é resolvido por todos os alunos, mas o número e a dificuldade dos exercícios resolvidos de seguida podem ser adaptados ao nível de cada aluno. Terceiro, o facto de existir um contacto mais prolongado (4.5 horas semanais) com um único docente num turno de pequena dimensão permite criar um relação de maior proximidade e confiança entre o docente e os alunos. Esta proximidade tem um forte impacto na participação, assiduidade e percentagem de desistências dos alunos.

As aulas práticas laboratoriais são uma componente importante, porque permitem que os alunos consolidar os conceitos apreendidos e exercitados nas aulas teórico-práticas com a resolução de exercícios complementares.

No mini-projeto final os alunos desenvolvem uma aplicação simples, o que permite aos alunos aplicarem toda a matéria apreendida durante as aulas. O apoio ao projeto durante as aulas permite, não só, resolver dificuldades pontuais dos alunos, como também ajudar os alunos a melhorar a organização do programa e a qualidade do código escrito.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Lectures are taught to small classes (around 28 students) and it includes theoretical exposition and practical exercises. This organisation has several advantages. First, students can apply immediately the concepts being explained. Therefore, the teacher can right away address any student's questions or doubts. Second, it supports coordinated teaching of students with varying levels of programming knowledge. This variability is tackled by having, for each relevant concept, a collection of exercises with different difficulty levels. Generally, after a theoretical exposition, all students must solve a core exercise that applies that concept. However, the exercises to be solved after the completion of the core exercise can be adapted to the student's knowledge. Third, the fact that each class has a single teacher for a small number of students it creates a close relationship between the students and the teacher, increasing class participation and attendance and decreasing the number of students dropping out of the course.

In the lab sessions students solve complementary exercises to consolidate the concepts learned and practiced during lectures.

In the final mini-project students have to develop a simple application that covers all concepts learned through the course. Project support during classes, not only addresses student's particular difficulties, but more importantly helps the students improve their program design and code quality.

3.3.9. Bibliografia principal:

- Cay Horstmann, *Java Concepts, 6th edition, Wiley, 2009.*
- Walter Savitch, *Java: An Introduction to Computer Science and Programming, 4th edition, Prentice-Hall, 2005.*
- David J. Eck, *Programming Using Java, Online book, <http://math.hws.edu/javanotes>, 2004.*

Mapa IV - Sistemas Lógicos / Digital Systems

3.3.1. Unidade curricular:

Sistemas Lógicos / Digital Systems**3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:**

Luis Filipe dos Santos Gomes (Responsável e Regente) (T-42h; OT-2h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Pedro Miguel Negrão Maló (P-126h; OT-2h)

Anikó Katalin Horváth da Costa (P-42h; OT-2h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Descrever sistemas digitais combinatórios através de expressões algébricas booleanas, tabelas de verdade e esquemáticos.

Aplicar metodologia de síntese de circuitos combinatórios.

Converter números entre diferentes bases de numeração, tais como decimal, binário, hexadecimal e octal.

Analisar métodos de decomposição modular de circuitos combinatórios, incluindo circuitos de aritmética binária.

Aplicar técnicas expeditas de desenho de contadores.

Aplicar metodologia de síntese de máquinas de estados síncronas, partindo de diagramas de estado.

Realizar sistemas digitais de reduzida/média complexidade através da sua decomposição em parte de dados e de controlo.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Describe digital systems through combinatorial Boolean algebraic expressions, truth tables, and schematics.

Apply methodology for synthesis of combinational circuits.

Convert numbers between different base numbering, such as decimal, binary, hexadecimal and octal.

Analyze methods of modular decomposition of combinational circuits, including circuits for binary arithmetic.

Apply techniques for expeditious design of counters.

Apply methodology for synthesis of synchronous state machines, starting from state diagrams.

Implement digital systems with low/medium complexity using its decomposition into control and data parts.

3.3.5. Conteúdos programáticos:

• *Álgebra de Boole: Postulados e teoremas da Álgebra de Boole; Tabelas de verdade*

• *Funções lógicas: Formas canónicas; Simplificação de funções; Mapas de Karnaugh; Método de Quine-McCluskey*

• *Sistemas de numeração: Conversão entre bases de numeração*

• *Aritmética binária: Soma e subtração; Complementos para 2 e para 1; Multiplicação e divisão.*

• *Circuitos combinatórios elementares: Comparadores; Codificadores e descodificadores; Conversores de código; "Multiplexers" e "demultiplexers".*

• *Elementos de memória biestáveis: Conceitos de "latch" e "flip-flop"; Biestáveis JK, D e T.*

• *Circuitos sequenciais: Noção de sistema síncrono e assíncrono; Registos; Desenho expedito de contadores.*

• *Máquinas de estado síncronas: Diagramas de estado, Circuitos Moore e Mealy, Síntese.*

• *Dispositivos específicos: Memórias; Dispositivos de lógica programável.*

• *Introdução a arquiteturas de transferência entre registos: decomposição em partes de controlo e de dados; introdução aos microprocessadores.*

3.3.5. Syllabus:

• *Boolean algebra: Theorems; Truth tables.*

• *Logical functions: Canonical representations; Function minimization; Karnaugh maps; Quine-McCluskey method.*

• *Numerical systems: conversions.*

• *Binary arithmetic: Addition and Subtraction; Two's-Complement; One's-Complement; Multiplication and division.*

• *Basic Combinatorial Circuits: Comparators; Encoders and decoders; Code Converters; Multiplexers and demultiplexers.*

• *Memory elements: Concepts of latch and flip-flop; Flip-flop JK, D and T.*

• *Sequential circuits: Synchronous and asynchronous circuits' concepts; Registers. Counter design.*

• *Synchronous state machines: State diagrams; Moore and Mealy circuits; Synthesis.*

• *Specific devices: Memories; Programmable logic devices.*

• *Introduction to register transfer architectures: decomposition in control and data parts; introduction to*

microprocessors.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os objetivos de aprendizagem indicados e a sequência dos conteúdos programáticos previstos estão diretamente associados, permitindo ir adquirindo gradualmente competências intermédias que suportarão o atingir do último objetivo referido, de elevada relevância para um engenheiro informático, uma vez que permite deixar clara a visão hardware das arquiteturas computacionais (computadores e outros equipamentos eletrónicos) que serão posteriormente (e regularmente) utilizadas pelos alunos (e futuros profissionais de engenharia).

Os conteúdos programáticos previstos encontram-se estruturados em quatro grupos, nomeadamente conceitos introdutórios (onde se incluem a álgebra de Boole e as funções lógicas), análise de módulos combinatórios (onde se incluem a aritmética binária e a análise de módulos combinatórios), análise de circuitos sequenciais (onde se incluem os elementos de memória, circuitos sequenciais e máquinas de estados síncronas) e análise de circuitos digitais de reduzida/média complexidade obtidos através da sua decomposição em parte de controlo e parte de dados (onde se integram os conhecimentos e competências adquiridos anteriormente).

Os objetivos de aprendizagem identificados permitem ir verificando a obtenção de competências (ao nível do saber, do saber fazer e de soft-skills) ao longo dos conteúdos apresentados, sendo possível uma associação direta com os referidos grupos de conteúdos programáticos.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The foreseen learning outcomes and the provided topics sequence of the syllabus are directly associated, allowing gradually acquiring intermediate skills that will support the last referred goal. This is of high relevance for a computer engineer, since it allows to clarify the hardware vision of the computing architectures (computers and other electronic equipment) that will later be regularly used by students (and future engineering professionals).

The provided syllabus are structured into four groups, including introductory concepts (which include Boolean algebra and logic functions), combinatorial analysis modules (which include binary arithmetic and analysis of combinatorial modules), circuit analysis sequential (which include memory elements, sequential circuits and synchronous state machines) and small / medium complexity digital circuit analysis obtained by its decomposition into control and data parts (which incorporate the knowledge and skills previously acquired). The identified learning outcomes allow checking achievement of specific skills (at different levels as knowledge, know-how and soft-skills), as it is possible a direct association between learning outcomes and those groups within the proposed syllabus.

3.3.7. Metodologias de ensino (avaliação incluída):

Consideram-se 2 aulas teóricas de 1h30min cada (3 horas/sem.) e uma aula prática (3 horas/sem.).

As aulas teóricas são aulas de exposição onde se fomenta a discussão de temas, permitindo dar ênfase diferenciada em aspetos conceptuais, bem como tecnológicos.

As aulas práticas são aulas de laboratório, permitindo dar ênfase diferenciada em vários aspetos, nomeadamente a resolução de exercícios, a utilização de ferramentas computacionais de simulação e síntese de sistemas digitais, e a experimentação física através da implementação de circuitos digitais utilizando circuitos elementares discretos, bem como dispositivos de lógica programável (FPGAs). Cada grupo de trabalho recebe uma placa de experimentação (com uma FPGA) permitindo a experimentação fora do laboratório. A avaliação é garantida através de 2 testes individuais na componente teórica e a componente prática através de um trabalho individual (1ª metade do semestre) e de um trabalho de grupo a apresentar no final do semestre.

3.3.7. Teaching methodologies (including evaluation):

Contact with students is accomplished in two lectures of 1h30min each (3 hours / wk.) and one lab class (3 hours / week).

The lectures are classes where exposure promotes discussion of topics, allowing emphasizing different aspects at conceptual and technological levels.

The laboratory classes allow integration of different emphasis on several aspects, such as problem solving, the use of computational tools for simulation and synthesis of digital systems, and physical experimentation by implementing digital circuits using discrete elementary circuitry as well as programmable logic devices (FPGAs). Each group receives one experimentation board (with one FPGA) allowing testing outside the laboratory.

The evaluation is ensured by two individual tests in theoretical component, and as practical component through an individual work (first half of the semester) and a small-project to be presented by each group of students at the end of the semester.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Complementando a aquisição de conhecimentos, aptidões e competências a desenvolver pelos estudantes ao nível conceptual, é dada uma ênfase especial ao nível da experimentação e da utilização de tecnologias de implementação de circuitos digitais, permitindo reduzir a distância normalmente observada nos estudantes quando se trata de “mexer” diretamente com dispositivos físicos. Para isso todos os grupos de trabalho (constituídos tipicamente por três estudantes) recebem um kit de experimentação constituído por um dispositivo lógico programável de complexidade média (uma FPGA), sendo possível a sua utilização fora do laboratório de aulas e sua integração nos processos de estudo autónomo dos estudantes. Desta forma, a resolução analítica de problemas propostos é complementada com a experimentação associada, permitindo aumentar os níveis de sucesso na aprendizagem.

Assim, os objetivos de aprendizagem indicados são plenamente suportados pela metodologia de ensino proposta.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Complementing the acquisition of knowledge, skills and competencies developed by students at the conceptual level, a special emphasis is given to the experimentation level, as well as to the use of digital circuitry implementation technologies, reducing the distance usually observed in students when it comes to "play" directly with physical devices. For that, all groups of students (typically composed of three students) receive an experimentation kit equipped with a programmable logic device of medium complexity (an FPGA), which can be used outside of the lab classes, and be completely integrated in the processes of students' autonomous study. In this way, the analytical resolution of problems is complemented with associated experimentation, enabling improving levels of success in the learning process.

Thus, the learning objectives listed are fully supported by the teaching methodology proposed.

3.3.9. Bibliografia principal:

1. *Digital Logic Circuit Analysis & Design - Victor P. Nelson, H. Troy Nagle, J. David Irwin, Bill D. Carroll - Prentice Hall - ISBN 0-13-463894-8*

2. *Digital Design: Principles and Practice - John F. Wakerly - Prentice-Hall - ISBN 0-13-082599-9*

3. *Logic and Computer Design Fundamentals - M. Morris Mand, Charles Kime - Prentice-Hall - ISBN 0-13-182098-2*

Mapa IV - Competências Transversais para Ciências e Tecnologia/ Soft Skills for Science and Technology

3.3.1. Unidade curricular:

Competências Transversais para Ciências e Tecnologia/ Soft Skills for Science and Technology

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Ruy Araújo da Costa (Responsável e Regente) (TP – 10h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

José Carlos Ribeiro Kullberg (P – 50h)

José Manuel Matos Ribeiro da Fonseca (P – 50h)

Nelson Fernando Chibeles Pereira Martins (P – 50h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

No final da UC um aluno deve ser capaz de:

-escrever o seu Curriculum Vitae (CV) e preparar-se para uma entrevista profissional;

-perceber a importância do desenvolvimento programado de atividades que contribuam para o enriquecimento do seu CV ao longo do tempo;

-perceber a importância dos Testes Psicotécnicos no acesso ao mercado de trabalho;

-perceber a importância do domínio básico da Língua Inglesa na área de Ciências e Tecnologia(CT);

-comunicar por escrito de modo adequado na área de CT;

-preparar uma apresentação oral, apoiada por PowerPoint,na área de CT;

-utilizar folhas de cálculo Excel produzindo gráficos com facilidade;

-utilizar no Excel o Solver e ser capaz de programar funções em Visual Basic;

-pesquisar Bibliografia através de bases de dados referenciais ou motores de pesquisa generalistas e analisar Informação,tendo presente exigências de ordem ética e deontológica;

- gerir adequadamente o tempo e trabalhar em equipa;
- compreender a importância da liderança.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

After this curricular unit, any student should be able to:

- write his (her) Curriculum Vitae and prepare for a job interview;
- understand the importance of taking steps to make his (her) Curriculum Vitae more appealing;
- understand how important Psychometric Testing is when accessing the job market;
- understand how important English is in the Science and Technology area;
- write an essay in the Science and Technology area;
- prepare an oral presentation in a Science and Technology topic, using PowerPoint;
- use Excel spreadsheets and be able to represent data in graphs;
- use Excel's Solver and be able to program functions in Visual Basic;
- carry out bibliographic research using referential databases or generic search engines, and critical analysis of scientific information considering both ethical and deontological issues;
- manage time adequately and be able to carry out team work effectively;
- understand the importance of leadership.

3.3.5. Conteúdos programáticos:

- 1 - Curriculum Vitae, Entrevista e Testes Psicotécnicos.
- 2 - Comunicação em Ciências e Tecnologia.
- 3 - Utilização avançada de folhas de cálculo Excel.
- 4 - Pesquisa bibliográfica e análise de informação. Ética e deontologia.
- 5 - Gestão do tempo. Trabalho de equipa. Liderança.

3.3.5. Syllabus:

- 1 - Curriculum Vitae, Job interview and Psychometric testing.
- 2 - Communicating in Science and Technology.
- 3 - Advanced use of Excel spreadsheets.
- 4 - Bibliographic research and critical analysis of scientific information.
- 5 - Time management, team work and leadership.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A unidade curricular (UC) visa dotar os alunos das competências consideradas essenciais para a sua progressão ao longo de um curso na área de Ciências e Tecnologia e sua posterior integração no mercado de trabalho.

Para motivar os alunos, cada um dos 5 temas é apresentado de modo “invulgar”, permitindo-lhes constatar as suas naturais fraquezas e motivando-os para os conteúdos da UC.

Cada tema é abordado numa semana de aulas, visando preparar o aluno para:

- a entrada no mercado de trabalho através da elaboração do seu CV e para as entrevistas e testes psicotécnicos;
- preparar e efetuar uma apresentação científica, que lhe será útil quer no seu percurso académico quer na sua vida profissional;
- utilizar o Excel como ferramenta de cálculo de uso geral em diferentes contextos;
- pesquisar e selecionar informação científica e técnica de forma a fundamentar corretamente os trabalhos que efetua;
- gerir adequadamente o seu tempo e trabalhar em grupo, reconhecendo a importância da liderança.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

In this curricular unit students are exposed to soft skills deemed important to their progress in a Science and Technology course and in their future jobs.

To get the students attentions, each of the five topics in this unit is introduced in an “unusual” way, allowing them to grasp their natural weaknesses and motivating them for the topics potential.

Each theme is worked throughout one week, preparing the students to:

- deal with CV writing, job interviews and psychometric testing;
- write an essay or make an oral presentation in a Science and Technology topic, which will be useful throughout their University curricula as well as in a job;
- use Excel as a general calculus tool in different contexts;
- know how to search and select scientific and technical information, thus being able to carry out sound work;

-adequately manage time, carry out group work and understand the importance of leadership.

3.3.7. Metodologias de ensino (avaliação incluída):

Em cada semana será abordado um novo tema, que será explorado com uma abordagem idêntica:

- À 2ª feira decorre uma sessão prática de 2h com uma tarefa inicial curta, que expõe os alunos à relevância do tema;*
- À 3ª e 4ª feiras decorrem duas sessões práticas de 4h cada, com tarefas mais complexas que deverão ser desenvolvidas na aula e fora da aula e que envolverão apresentações orais, com ou sem suporte informático. Os docentes farão críticas construtivas aos trabalhos desenvolvidos pelos alunos, enquadrando-os no tema;*
- À 5ª feira decorre uma sessão teórico-prática de 2h onde são apresentados os aspetos fundamentais do tema, destacados os erros a evitar durante a exploração dos conteúdos do tema e realçadas as principais ferramentas que podem ser utilizadas.*

A avaliação final da u.c. será baseada no trabalho desenvolvido individualmente e em grupo durante cada semana e em testes individuais executados na plataforma de e-learning moodle em ambiente controlado.

3.3.7. Teaching methodologies (including evaluation):

In each week a new theme will be developed. The general approach for every theme is similar:

- on Mondays a 2h practical session takes place: students are requested to perform a short task that will reveal the importance of the theme;*
 - on Tuesdays and Wednesdays two 4h practical sessions take place: students have to develop a more complex task and have to make an oral presentation, in which they may use PowerPoint. Teachers will make comments and critiques to the students' work;*
 - on Thursdays a 2h theoretical-practical session is used to present the theme's fundamentals, the most common mistakes to be avoided and the main tools that can be used during the theme's exploration.*
- Assessment of this course takes into account both the weekly individual and group work, as well as tests carried out in moodle e-learning platform, in a controlled environment.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

1 - Na 2ªfeira solicita-se ao aluno que escreva o seu Curriculum Vitae (CV) atual, com vista a uma candidatura virtual a uma bolsa, ou um pequeno emprego na Biblioteca da Faculdade. Em seguida, discute-se os conteúdos alternativos de um CV e formas de apresentação. Solicita-se que os alunos compareçam na 3ªfeira em "modo de entrevista" para um emprego, com o seu CV. Seleciona-se alguns alunos e procede-se a entrevistas simuladas. Comenta-se os vários aspetos relevantes (p.ex., CV, vestuário, apresentação, dicção). Na 4ªfeira, solicita-se ao aluno que imagine o seu CV daí a 5 ou 6 anos e o escreva, com vista a uma candidatura a um emprego, pós-Mestrado. Solicita-se a reflexão sobre a evolução dos dois CV's e sobre a importância do desenvolvimento programado de atividades que contribuam para o enriquecimento do CV ao longo do tempo. Os alunos são ainda testados, via moodle, com Testes Psicotécnicos e na sessão de 5ªfeira chama-se a sua atenção para a importância dos referidos Testes.

2 - Solicita-se que grupos de 4 alunos analisem um pequeno texto de divulgação na área de Ciências e Tecnologia (C&T), escrito em Inglês, retirado de uma revista internacional e que produzam um resumo escrito adequado em Português e preparem uma apresentação oral sobre o tema e eventuais extensões, apoiada por PowerPoint. São feitos comentários aos materiais produzidos e à apresentação oral. Assim, os alunos são sensibilizados para a importância do domínio básico da Língua Inglesa, obtendo ainda formação sobre a comunicação escrita e oral na área de C&T.

3 - Na 2ªfeira, solicita-se aos alunos que representem graficamente algumas funções associadas a diversas áreas de aplicação. Introduce-se a utilização do Excel no contexto da representação gráfica dessas funções. Na 3ªfeira apresenta-se a cada grupo um conjunto de folhas de cálculo com informações relativas a um mesmo grupo de indivíduos (uma folha para cada indicador). Solicita-se que criem uma folha de cálculo única com todas as informações disponíveis sobre cada indivíduo de um subgrupo do grupo inicial. Posteriormente, apresenta-se as funções de referência do Excel que permitem levar a cabo essa atividade de modo expedito. Na 4ªfeira solicita-se a determinação da solução de uma equação, ou a resolução de um problema, para introduzir o "Solver" do Excel. Introduce-se, ainda, o módulo de Visual Basic do Excel, com a escrita de funções específicas.

4 - Dado um tema, solicita-se a realização de pesquisa de Bibliografia. Discute-se os cuidados a ter na pesquisa bibliográfica e na análise da Informação. Destacam-se as exigências de ordem ética e deontológica, apresentando-se exemplos atuais e internacionais de figuras políticas de relevo envolvidas em situações de plágio e suas consequências.

5 - Aborda-se a Gestão do Tempo no contexto universitário e no contexto da Gestão de Projetos. Analisa-se as vantagens e desvantagens do trabalho em equipa. Analisa-se as características relevantes de um líder e a sua importância.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

- 1 - On Monday each student is asked to write his (her) present Curriculum Vitae (CV), to apply for a virtual scholarship, or a job at the campus Library. Afterwards, alternative contents of a CV are discussed, as well as different ways to present a CV. Students are requested to come on Tuesday on a "job interview mode" with their CVs. A few students are selected and job interviews are simulated. Different aspects are evaluated (e.g., CV; clothing, presentation, diction). On Wednesday each student is asked to imagine his(her) CV in 5 or 6 years and write it, applying for a job after completing the MSc course. Students have to reflect about the CV's evolution and realize that they should take steps to make their CVs more appealing. Using moodle e-learning platform, students carry out Psychometric Tests and on Thursday these testing is highlighted as an important step in a future job interview process.
- 2 - Small texts are selected in English language magazines, covering Science and Technology (S&T) topics. Each group of 4 students has to analyze one of those texts, make a written summary in Portuguese and prepare an oral presentation of the theme and eventual extensions, using PowerPoint. Comments will be made both to the written summary and to the presentation. Thus, students realize the importance of using English and acquire skills in written and oral presentations in the ST area.
- 3 - On Monday, students are requested to draw graphs of functions associated with different areas of application. Excel is introduced as an easy means of drawing those graphs. On Tuesdays each group of students receives a set of spreadsheets regarding a set of individuals (each sheet for a different indicator). Students are requested to produce one spreadsheet for a given subset of individuals, with all information regarding all indicators. Afterwards, lookup and reference Excel functions are presented as a way to carry out that task quickly. On Wednesday students are requested to derive the solution of an equation, or to solve a problem, and Excel's "Solver" is introduced. Excel's Visual Basic module is presented and students are taught to write custom-made functions.
- 4 - Given a theme, students are requested to carry out a bibliographic research. Students are instructed to be careful when retrieving and analyzing information. Ethical and deontological demands are presented. Recent international and prominent examples of fraud and their consequences are presented.
- 5 - Time Management is addressed in a university context as well as in a Project Management context. Advantages and disadvantages of group work are analyzed. Leader's characteristics are addressed, as well as the importance of leadership.

3.3.9. Bibliografia principal:

- Costa, R., Kullberg, J., Fonseca, J., Martins; N., "Manual de Competências Transversais para Ciências e Tecnologia – FCT/UNL" (2012) – em elaboração / in preparation.

Mapa IV - Análise Matemática II E / Mathematical Analysis II E

3.3.1. Unidade curricular:

Análise Matemática II E / Mathematical Analysis II E

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Maria Fernanda Alves da Veiga de Oliveira (responsável e regente) (T-84h; P-84h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

António José Mesquita da Cunha Machado Malheiro (P-126h)

Ana Maria Manteigas Pedro (P-126h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Identificar e resolver problemas de valores iniciais p/ algumas equações diferenciais de 1ª ordem e implementar um método numérico de aproximação—método de Euler; fazer o estudo geométrico de curvas e de quádras incluindo uso de diferentes coordenadas; estudar a continuidade e diferenciabilidade de funções de várias variáveis (incluindo os casos de indeterminação) aplicando noções topológicas, calcular aproximações lineares locais, e interpretar geometricamente estas noções, determinando planos tangentes a superfícies definidas quer explícita quer implicitamente e conhecendo e aplicando as propriedades geométricas do vetor gradiente; derivar funções compostas e definidas implicitamente; determinar extremos relativos de funções de várias variáveis; determinar extremos condicionados através do método dos multiplicadores de Lagrange; calcular integrais múltiplos usando o teorema de Fubini e usando mudança de variáveis; calcular áreas e volumes de superfícies e sólidos usando integrais múltiplos.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Recognize and solve initial value problems for linear and separable 1st order differential equations and formulate a numerical approximation method (Euler's method); study plane and 3D curves and quadric surface, including use of different coordinate systems; study the continuity and differentiability of 2 and 3 variable functions (including solving indeterminations) applying topological notions and calculate linear local approximations, and master the geometrical interpretation of these notions, e.g. determining tangent planes to explicitly or implicitly defined surfaces and know and apply the geometrical properties of the gradient; use the chain rule and the rule to calculate partial derivatives of implicitly defined functions; determine local extrema of two variable functions; determine conditional extrema using Lagrange multipliers method; calculate multiple integrals using Fubini's theorem and change of variables; calculate surface areas and volume of solids using multiple integrals.

3.3.5. Conteúdos programáticos:

Equações diferenciais: noções gerais, estudo de equações de 1ª ordem lineares e separáveis, método de Euler. Espaços vetoriais de dimensão finita: quádricas, diferentes sistemas de coordenadas. Funções vetoriais: interpretação geométrica, noções da análise para funções vetoriais, interpretação geométrica da derivada – vetor tangente, reta tangente e plano normal a uma curva num ponto. Funções de várias variáveis – cálculo diferencial: noções topológicas em R^2 e R^3 , limites e continuidade, derivadas parciais e diferenciabilidade, regra de derivação da função composta, aplicação ao cálculo de derivadas parciais de funções definidas implicitamente, derivada direcional e propriedades geométricas do gradiente, plano tangente e reta normal a uma superfície num ponto, extremos locais, extremos condicionados – método dos multiplicadores de Lagrange. Integrais múltiplos: integrais duplos e triplos, mudança de variáveis, aplicação ao cálculo de áreas de superfícies e volumes de sólidos.

3.3.5. Syllabus:

1. *Differential equations: general notions, 1st order separable and linear equations, Euler's method*
2. *Finite dimensional linear spaces: quadric surfaces, polar, cylindrical and spherical coordinates*
3. *Vector-valued functions: geometrical interpretation, differential and integral calculus, geometrical interpretation of the derivative*
4. *Functions of several variables: topological notions in R^2 and R^3 , limits and continuity, partial derivatives and differentiability, chain rule and application to the calculation of partial derivatives of implicitly defined functions, directional derivative and geometrical properties of the gradient, tangent plane, local extrema, conditional extrema – Lagrange multipliers*
5. *Multiple integrals: double and triple integrals, change of variables in multiple integrals, applications of integrals*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Para que os objetivos referidos sejam alcançados é necessário o conteúdo programático referido. Por outro lado, as matérias indicadas no conteúdo programático são suficientes para que o aluno cumpra todos os objetivos.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The program contents are necessary for the achievement of the objectives. On the other hand, the materials indicated in the syllabus are sufficient for the meeting of the objectives by the student.

3.3.7. Metodologias de ensino (avaliação incluída):

O ensino consiste na leção de aulas teóricas, onde é apresentada e explicada toda a matéria referida nos conteúdos programáticos. São fornecidas fichas de exercícios aos alunos para serem resolvidos nas aulas práticas, com o conhecimento adquirido previamente nas aulas teóricas.

Os alunos dispõem ainda do designado horário de dúvidas onde podem esclarecer as suas dúvidas com o professor

É necessária a obtenção de frequência para poder realizar as provas de avaliação. Segundo a norma vigente na Faculdade, a avaliação é contínua, consistindo esta em quatro testes a realizar ao longo do semestre, ou então é por um exame final.

3.3.7. Teaching methodologies (including evaluation):

The professor teaches the course by lectures, where he explains all the topics that constitute the syllabus. A problem list is provided to students to be solved in practical sessions, applying prior knowledge acquired in the lectures.

Students can also meet their professor in extra hours to clarify points where they feel difficulties.

Admission to evaluation tests requires “frequency”.

According to Faculty regulations, approval can be obtained during class period, through 4 partial tests, or alternatively through a final exam after the end of class period.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os métodos de ensino permitem aos alunos apreender todos os conteúdos programáticos. Visam uma profunda compreensão das noções teóricas e o domínio da sua manipulação e aplicações, nomeadamente através da resolução de problemas.

Assim, os estudantes ficam aptos para atingir todos os objetivos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The teaching methods presented allow the students to achieve all the syllabus contents. Their aim is to provide a deep understanding of theoretical concepts and the mastering of their manipulation and application namely by problem solving training.

In this way students become able to achieve all objectives.

3.3.9. Bibliografia principal:

- 1. Ana ALVES de SÁ, Cálculo Diferencial em R^n Exercícios resolvidos, FCT/UNL, 2010*
- 2. Ana ALVES de SÁ, Bento LOURO, Cálculo Diferencial em R^n , FCT/UNL, 2010*
- 3. Howard ANTON, Irl BIVENS, Stephen DAVIS, Calculus, 8th edition, Wiley (New Jersey), 2005*
- 4. Tom APOSTOL, Calculo, Vol.2, Reverté, 1993.*
- 5. SALAS, HILLE, Calculus, one and several variables, John Wiley and Sons, Inc, 1995.*
- 6. STEWART, J., Calculus, 3ª edição, Brooks/Cole Publishing Company, 1995.*
- 7. EARL W. SWOKOWSKI, Cálculo com geometria analítica, Vol. 2, MacGraw-Hill,1983.*

Mapa IV - Matemática Discreta / Discrete Mathematics

3.3.1. Unidade curricular:

Matemática Discreta / Discrete Mathematics

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Vítor Hugo Bento Dias Fernandes (apenas responsável, não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Isabel Maria Oitavem Fonseca da Rocha Kahle (Regente) (T-42h; P-56h; OT-3h)

Maria de Fátima Vale de Gato Santos Rodrigues (P-112h; OT-3h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Pretende-se que os alunos adquiram conceitos básicos em Teoria de Grafos e no campo dos Fundamentos da Matemática, nomeadamente no que se refere a Conjuntos e Aplicações. Pretende-se também que os alunos dominem algumas técnicas de demonstração em cada uma das áreas referidas e que saibam aplicar resultados e algoritmos na resolução de problemas.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The student is supposed acquire basic knowledge on Graph Theory, Set Theory and Number theory, in learning process, logical reasoning and critical mind are developed.

3.3.5. Conteúdos programáticos:

Parte 1 – Conjuntos, relações e funções

1. *Conjuntos: representações e operações básicas; conjunto das partes; cardinalidade*
2. *Relações binárias: equivalências e ordens parciais*
3. *Funções: bijeções; inversão e composição*
4. *Aritmética modular e congruências lineares*

Parte 2 – Indução

1. *Definições indutivas*
2. *Indução nos naturais e estrutural*
3. *Primeiro e segundo princípios de indução*
4. *Funções recursivas e provas por indução*

Parte 3 – Contagem

1. *Cálculo de somatórios, permutações e combinações*
2. *Relações de recorrência e Teorema Master*

Parte 4 – Grafos e Aplicações

1. *Generalidades*
2. *Conexidade*
3. *Árvores*
4. *Grafos Eulerianos*
5. *Matrizes e grafos*

3.3.5. Syllabus:

Part 1 – Sets, relations and functions

1. *Sets: representations and basic operations; power set; cardinality*
2. *Binary relations: equivalence and partial orders*
3. *Functions: bijections; composition and inverse*
4. *Modular arithmetic and linear congruences*

Part 2 – Induction

1. *Inductive definitions*
2. *Induction over natural numbers and structural induction*
3. *Complete induction and course-of-values induction*
4. *Recursive functions and proofs by induction*

Part 3 – Counting

1. *Sums, permutations and combinatorics*
2. *Recurrence relations and Master theorem*

Part 4 – Graphs and applications

1. *Introduction*
2. *Connexity*
3. *Trees*
4. *Euler graphs*
5. *Matrices and graphs*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos programáticos estão divididos em quatro partes. A parte, 2 para além de contribuir para o desenvolvimento do raciocínio lógico, introduz os métodos básicos de demonstração em contextos discretos. As partes 1, 3 e 4 versam, respectivamente, sobre aspectos elementares de teoria de conjuntos, teoria de números e teoria de grafos.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The content of this curricular unit is divided in four parts. The main contributions of part 2 consist in the development of logic reasoning skills and the introduction to the basic proof methods used in discrete contexts . Part 1, 3 and 4 are devoted to basic issues of set theory, number theory and graph theory, respectively.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas práticas serão feitos exercícios de aplicação dos conceitos apresentados nas aulas teóricas. Os alunos dispõem dos enunciados dos exercícios utilizados nas aulas práticas, bem como de elementos de apoio às aulas teóricas.

A assistência às aulas teóricas é considerada fundamental para a realização com sucesso da disciplina (não é obrigatória apenas por questões logísticas). A assistência às aulas práticas é obrigatória.

Estando os conteúdos programáticos da disciplina organizados em quatro partes, no final de cada parte será efectuado um mini-teste que avaliará a aquisição de conhecimentos referentes a essa parte da matéria. Os mini-testes serão realizados nas aulas teóricas (se o número de alunos assim o permitir) e permitem obter aprovação à disciplina ou dispensar a partes da matéria no exame final, consoante o aluno tenha obtido classificação positiva em todos os mini-testes ou apenas a alguns deles.

3.3.7. Teaching methodologies (including evaluation):

Basic concepts will be introduced in lectures ("aulas teóricas") and problems will be solved in exercise classes ("aulas práticas"). The course is organized in four parts, therefore there will be four mini-tests. Positive grades in some of the mini-tests allow students to skip those parts in the final exam. Positive grades in all mini-tests leads to approval.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A metodologia adoptada combina uma vertente teórica, de cariz mais expositivo, com uma vertente prática onde o estudante é chamado a participar ativamente. Associando assim ao conhecimento adquirido, a capacidade do aplicar, estimulando em particular a capacidade de raciocínio.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The adopted methodologies combine theoretical and exercise classes. The exercise classes call the students to take active part of their own learning process. The exercises are designed to guide the students to a deeper understanding of the taught topics, and to develop their reason ability.

3.3.9. Bibliografia principal:

[1] R. Johnsonbaugh, *Discrete Mathematics*, Prentice Hall International, 1997

[2] T. S. Blyth e E. F. Robertson, *Sets and Mappings*, Chapman and Hall, 1986

[3] N. L. Biggs, *Discrete Mathematics*, Oxford Science Publications, 1994

[4] K. A. Ross e C. R. B. Wright, *Discrete Mathematics*, Prentice Hall International, 1999

[5] R. J. Wilson e J. J. Watkins, *Graphs an Introductory Approach*, Wiley, 1990

[6] S. Lipschutz, *Set Theory and Related Topics*, Schaum's Outline Series, Mc Graw-Hill, 1964

[7] D.M. Cardoso, J. Szymanski e M. Rostami, *Matemática Discreta*, Escolar Editora, 2009

Mapa IV - Arquitetura de Computadores / Computer Architecture

3.3.1. Unidade curricular:

Arquitetura de Computadores / Computer Architecture

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Alberto Cardodo e Cunha (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Rodrigo Seromenho Miragaia Rodrigues (Regente) (T-84h; OT-2h)

Hervé Miguel Cordeiro Paulino (P-112h; OT-2h)

Maria Cecília Farias Lorga Gomes (P-28h; OT-2h)

Cármén Pires Morgado (P-56h; OT-2h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Aos profissionais das áreas da informática e tecnologias de informação exige-se a compreensão dos vários níveis de software e hardware que permitem que programas escritos numa linguagem de alto nível como C ou Java possa correr num sistema computacional. O objetivo central desta cadeira é fornecer uma visão global da organização de um computador, do desenho das camadas de software e hardware que formam um sistema computacional, e da interação entre as diferentes camadas, com ênfase particular nos aspetos da arquitetura de computadores.

Saber:

- *Organização geral do hardware de um sistema computacional.*
- *Representação no hardware de dados e instruções.*
- *Processo de tradução entre os programas em C, o assembly, e a linguagem máquina.*
- *Interface de baixo nível para os dispositivos de entradas/saídas.*
- *Características das unidades de memória.*

Saber Fazer:

- *Programação em C, assembly, e a tradução entre ambos.*
- *Desenhar os componentes básicos de um CPU.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Computer science and IT professionals must be able to understand the various layers of software and hardware that allow for their programs written in a high-level language like C or Java to run a computer system. The central goal of this course is to provide a holistic view on how a computer is organized, the design of the software and hardware layers that form a computer system, and the interaction between these layers, with a particular focus on the computer architecture aspects of this design.

Knowledge:

- *The general organization of computer hardware, and the inner workings of the CPU.*
- *The hardware representation of data and instructions.*
- *The translation process from C programs to assembly language and machine language.*
- *The low level interfaces to I/O devices.*
- *The organization and characteristics of memory units.*

Application:

- *C programming, assembly programming, translating between the two.*
- *Designing the basic components of a CPU.*

3.3.5. Conteúdos programáticos:

1. *Programação em C*
2. *Representação numérica*
3. *Linguagem assembly: tipos de instruções, formato das instruções, procedimentos e convenção de chamadas a procedimentos.*
4. *Compilação, ligação, assemblagem, e carregamento.*
5. *Caches e unidades memória. Introdução breve ao suporte de hardware para memória virtual.*
6. *Organização interna da unidade de processamento. Processador de ciclo único. Pipelines. Conceitos avançados do desenho de microprocessadores.*
7. *Entradas / saídas.*

3.3.5. Syllabus:

1. *Programming in C.*
2. *Number representation.*
3. *Assembly language: instruction types, instruction format, procedures and calling conventions.*
4. *Compiling, linking, assembling, and loading.*
5. *Caches and memory units. Introduction to the hardware support for virtual memory.*
6. *Internal organization of processing units. Single cycle processor. Pipelining. Advanced concepts.*
7. *I/O.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O conteúdo programático da cadeira fornece aos alunos uma visão integrada das diferentes camadas que

formam um sistema computacional, começando com as camadas de níveis de abstração superiores, e gradualmente explicando como é que os níveis imediatamente abaixo suportam essas mesmas abstrações. Em particular, o programa da cadeira permite aos alunos compreender a linguagem assembly, e de que forma este código é derivado do código C. Adicionalmente, os alunos compreenderão de que forma o assembly é traduzido para linguagem máquina, e qual o hardware que é necessário para suportar a execução de código máquina, focando em vários aspetos da organização de um processador como a unidade de processamento, a memória, ou as entradas/saídas.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The course syllabus provides students with a integrated view of the different layers that form a computer system, starting with the higher levels of abstraction, and gradually explaining how the layers immediately below provide the necessary support to implement these abstractions.

In particular, the course syllabus allows students to gain an understanding of assembly language, and how this code is derived from code written in C. Furthermore, the students will learn how assembly code is translated to machine language programs, and what is the hardware that is required to execute them, focusing on several aspects of computer organization like processing, memory, and I/O units.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas explicar-se-ão e discutir-se-ão os sucessivos tópicos do programa da cadeira. No último terço de cada aula proceder-se-á, sempre que possível, à resolução de um problema prático cuja solução requeira a aplicação dos conhecimentos aprendidos no início da aula.

Nas aulas práticas serão desenvolvidos trabalhos de programação, cujo objetivo é consolidar os conceitos que foram aprendidos nas aulas teóricas. Após concluírem estes trabalhos, os alunos devem discutir com os docentes o comportamento dos programas desenvolvidos e o seu encadeamento com os conceitos aprendidos ao longo da cadeira.

Componentes da avaliação

- Dois testes que versam os conhecimentos teóricos e práticos da cadeira.*
- Dois projetos de programação em linguagens correspondentes a diferentes níveis de abstração*
- Vários pequenos trabalhos de casa individuais que permitem exercitar tópicos não cobertos pelos projetos, e que permitem uma carga de estudo e trabalho mais uniforme ao longo do semestre.*

3.3.7. Teaching methodologies (including evaluation):

In lectures we will explain and discuss the sequence of points in the course program. In the last third of each lecture we will discuss, whenever possible, the resolution of a practical exercise whose solution requires applying the knowledge learned in the lecture.

During lab classes students will develop programming assignments, whose objective is to consolidate the concepts that were learned in lecture. After concluding each assignment, students should discuss with the instructor the behavior of the programs they developed and the respective connection to the concepts learned throughout the course.

Evaluation components

- Midterm and term quizzes, testing the knowledge of both theoretical and practical concepts.*
- Two programming projects using languages that correspond to different levels of abstraction.*
- Several small homework assignments that enable the students to work on topics that are not covered by the projects, allowing for spreading the course load more uniformly.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As metodologias de ensino têm como objetivo permitir a compreensão aprofundada do desenho das camadas de software e hardware que formam um sistema computacional, e da interação entre as diferentes camadas. Tal será conseguido através da uma integração cuidada entre os conhecimentos explicados na aula teórica, a resolução nas mesmas aulas de exercícios práticos, e a aprendizagem dos conceitos através de exercícios de programação, quer nas aulas práticas quer nos projetos e trabalhos de casa. Estes componentes contribuirão em simbiose para a consolidação do conhecimento das características dos mecanismos e abstrações fornecidos pelas diferentes camadas que formam um sistema computacional. Adicionalmente, alguns exercícios de programação permitirão inferir as características do hardware e realçar a forma como o conhecimento destas características podem beneficiar o desempenho do software. Finalmente, os exercícios dos testes focar-se-ão em conhecimentos aprendidos quer nas aulas teóricas quer nas aulas práticas e projetos, de forma a dar uma visão integrada dos conhecimentos teóricos e práticos da cadeira.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The goal of the teaching methodologies is to allow for a deep understanding of the design of the software and hardware layers that form a computer system, and the interaction among them. This goal will be achieved by a careful integration of the knowledge obtained in lecture both through exposition and through solving practical exercises, with the knowledge obtained through programming exercises in lab classes, projects, and homework assignments. These components will contribute symbiotically to the consolidation of the knowledge about the characteristics of the different layers that form a computer system. In addition, some programming exercises will allow for inferring the characteristics of the hardware components and will highlight how the knowledge of these hardware characteristics can allow for building faster software. Finally, the questions that comprise the quizzes will focus on knowledge learned both through lectures and through the lab and project components of the course, thus enabling an integrated view of the practical and theoretical concepts learned in the course.

3.3.9. Bibliografia principal:

Bibliografia de base / Course reading

• David A. Patterson, John L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface (revised Fourth Edition)*. Morgan Kaufmann, 2011.

• Brian W. Kernighan, Dennis M. Ritchie. *C Programming Language (2nd Edition)*. Prentice Hall, 1988.

Bibliografia complementar / Complementary reading

• José Monteiro, Guilherme Arroz, Arlindo Oliveira. *Arquitetura de Computadores - dos Sistemas Digitais aos Microprocessadores*. IST Press. 2ª edição, 2009

• José Delgado e Carlos Ribeiro, *Arquitetura de Computadores, 2009, LTC Editora*.

Mapa IV - Programação Orientada Pelos Objetos / Object Oriented Programming

3.3.1. Unidade curricular:

Programação Orientada Pelos Objetos / Object Oriented Programming

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luis Manuel da Costa Caires (apenas Responsável não tem horas de contato)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Carla Maria Gonçalves Ferreira (Regente) (T-42h),

Adriano Martins Lopes (T-42h; P-56h)

Miguel Carlos Pacheco Afonso Goulão (P-56h)

Miguel Jorge Tavares Pessoa Monteiro (P-84h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

• *Conceitos fundamentais de Programação Orientada pelos Objetos (POO), como o de classe, interface, polimorfia e herança.*

• *Técnicas e algoritmos para o processamento de listas, cadeias de caracteres, vetores, dicionários e conjuntos, etc.*

• *Programação genérica.*

• *Bibliotecas de classes e sua utilização.*

• *Conceitos básicos de Engenharia de Software na perspectiva de um programador, como a qualidade, reutilização, custos e manutenção, ou eficiência de implementações.*

Saber fazer:

• *Projetar e Desenvolver aplicações com POO.*

• *Resolver problemas com POO.*

• *Usar abordagens de verificação e validação.*

• *Realizar de forma progressivamente mais autónoma, individualmente e em grupo, projetos de desenvolvimento de software.*

Competências Complementares:

• *Gosto pela programação e pela resolução de problemas.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Object-Oriented Programming (OOP) concepts, namely classes, interfaces, polymorphism and inheritance.*
- *Techniques and algorithms for processing data structures such as lists, strings, vectors, dictionaries, and sets.*
- *Basic mechanisms of generic programming.*
- *Class libraries and learn how to reuse them.*
- *Basic Software Engineering concepts, from a developer perspective, such as software quality, reuse, development and maintenance costs, or implementation efficiency.*

Application:

- *Design and develop software applications using OOP.*
- *Problem solving with OOP.*
- *Using verification and validation techniques.*
- *Increasingly autonomous software development skills, both in solo and as a team member.*

Soft-Skills:

- *Build up an interest for programming and solving problems.*

3.3.5. Conteúdos programáticos:

1. *Especificação de interfaces e respetiva documentação*
2. *Conceção de programas estruturados em classes*
3. *Polimorfia de interfaces*
4. *Herança de classes*
5. *Polimorfia de herança*
6. *Asserções*
7. *Testes*
8. *Tratamento de exceções*
9. *Programação genérica*
10. *Utilização de tipos genéricos e coleções*

3.3.5. Syllabus:

1. *Interface specification and documentation*
2. *Software design with classes*
3. *Interface polymorphism*
4. *Class inheritance*
5. *Inheritance polymorphism*
6. *Assertions*
7. *Unit and regression testing*
8. *Exception handling*
9. *Generic programming*
10. *Usage of collection generic types*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O programa incide sobre o desenvolvimento de software usando programação orientada pelos objetos. São estudados conceitos fundamentais deste paradigma, tais como classes, interfaces, polimorfia e herança. Estudam-se vários tipos de coleções, construídas com tipos genéricos, que armazenam e permitem manipular objetos em diferentes estruturas de dados. O estudo de técnicas de processamento destas estruturas permite exercitar a algoritmia, usando de forma eficiente bibliotecas de classes disponíveis. A qualidade do software desenvolvido é reforçada pela utilização de técnicas como a definição de asserções, a construção sistemática de testes, ou o tratamento de exceções. A construção de software reutilizável, com as vantagens que isso traz em termos de custos de desenvolvimento e manutenção é inculcida de modo transversal ao longo de toda a unidade curricular, e reforçada pela utilização de bibliotecas de classes genéricas.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus focuses on Object-Oriented software development. The main concepts of this paradigm, such as classes, interfaces, polymorphism and inheritance, are studied. Several types of collections that allow storing and manipulating of their elements in different data structures are also addressed. These data structures are built using generic data types. Learning techniques on how to efficiently reuse these collections, available from library classes, is used for developing the student's algorithmic skills. Software quality is fostered through the usage of techniques such as, specification of assertions, systematic construction of test batteries, and exception handling. The concern for building reusable software, with its underlying advantages with respect to development and maintenance costs, cross-cuts the whole unit and is reinforced with the reuse of generic classes libraries.

3.3.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas consistem na exposição de matéria, ilustrada com exemplos de aplicação, e suportada por diapositivos e pela utilização do ambiente de desenvolvimento no computador do docente.

Nas aulas de laboratório, os alunos resolvem exercícios que constituem exemplos concretos de aplicação dos conceitos dados nas anteriores aulas teóricas.

Os alunos podem esclarecer dúvidas durante as aulas ou nos horários de atendimento.

Avaliação:

- 2 trabalhos práticos, realizados em grupos de 2 alunos, valendo 20% da nota final cada um (NP).*
- 2 testes individuais, sem consulta, valendo 30% da nota final cada um, num total de 60% (NT).*
- Em época de recurso, a nota dos dois testes é substituída pela nota do exame, que vale 60% (NT) da nota final (NF).*

O aluno obtém aprovação se ambas as notas NP e NT forem superiores ou iguais a 9.5 valores. Esta regra aplica-se em todas as épocas de exame.

As melhorias de nota apenas se realizam por exame, sendo a nota final igual à nota do exame.

3.3.7. Teaching methodologies (including evaluation):

Lectures consist on a theoretical exposition, illustrated by application examples, and supported by slides and the usage of the software development environment of the lecturer.

In the lab sessions, students solve problems by concretely applying the concepts discussed in the previous lectures.

Students may clarify any doubts during classes or in the instructor's contact hours.

Evaluation:

- 2 practical assignments, in groups of 2 students. Each assignment is worth 20%, in a total of 40% (PG) of the final grade.*
- 2 individual tests, without consultation, each worth 30% of the final grade, in a total of 60% (TG) of the final grade.*
- In the appeal season, the tests grade is replaced by the exam grade, which is worth 60% (TG) of the final grade (FG).*

A student is approved if both PG and FG are greater or equal to 9.5 points. This rule applies in all evaluation seasons.

Grade improvement can only be done in the appeal exam, which is worth 100% of the final grade.

3.3.8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular:

Nas aulas teóricas a matéria é exposta e ilustrada em exemplos, normalmente relacionados com a resolução de problemas. Estes conceitos são depois exercitados nas aulas de laboratório, em que os alunos os utilizam na resolução de problemas concretos. Os dois trabalhos práticos de avaliação constituem peças fundamentais na consolidação dos conhecimentos teóricos apreendidos nas aulas e sua aplicação à resolução de problemas concretos. O primeiro incide sobre os conceitos fundamentais da programação orientada pelos objetos, enquanto que o segundo trabalho exercita também a reutilização de classes de biblioteca (em particular as relacionadas com coleções), tipos genéricos e das técnicas relacionadas com a qualidade do software desenvolvido. Os dois testes seguem uma distribuição de foco semelhante. A existência da avaliação intercalar constituída pelo primeiro projeto e o primeiro teste funciona como um incentivo forte para que os alunos acompanhem atempadamente a matéria lecionada na primeira metade do semestre, o que é fundamental para o sucesso na segunda metade do mesmo. A retroação dada ao longo da realização dos exercícios práticos e projetos, durante as aulas práticas, visa detetar e corrigir atempadamente as dificuldades que os alunos possam sentir na apreensão de conhecimentos. A realização de trabalhos em grupo e sua discussão com os docentes permite desenvolver os soft-skills enunciados na definição dos objetivos de aprendizagem.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The course topics are presented and illustrated through examples, usually related to solving problems, during the lectures. These concepts are further exercised in the labs, where students use them to solve concrete problems. The two practical projects carried out in the semester are essential for the consolidation of the knowledge acquired in the lectures and its application to solving concrete problems. The first project focuses on the basic concepts of object-oriented programming, while the second one also exercises the reuse of library classes (in particular those related to collections), generic types and techniques related to software quality. The two individual tests have a similar focus distribution. The existence of an evaluation in the middle of the semester, including the first project and the first test is a strong incentive for students to understand the topics discussed in the first half of the semester, which is essential for their success in the second half of the

semester. The feedback provided by the instructors to the exercises and projects, in the lab sessions, aims to detect and correct difficulties the students may feel, in a timely manner. The development of these group projects is also instrumental for students to build up the soft skills discussed while defining the learning objectives of this course.

3.3.9. Bibliografia principal:

Cay Horstmann, Java Concepts, 4ª Edição (ou seguintes), Wiley, 2005, ISBN-10: 0471697044.

Diapositivos de Programação Orientada pelos Objetos (serão disponibilizados gradualmente na página moodle da cadeira).

F. Mário Martins, Java 5 e Programação por Objetos, FCA, 2006, ISBN 978-972-722-548-4.

Mapa IV - Algoritmos e Estruturas de Dados /Algorithms and Data Structures

3.3.1. Unidade curricular:

Algoritmos e Estruturas de Dados /Algorithms and Data Structures

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel da Costa Caires (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Maria Armanda Simenta Rodrigues Grueau (Regente) (T-84h; OT-3h),

Vasco Miguel Moreira Amaral (P-84h; OT-3h)

Sofia Carmen Faria Maia Cavaco (P-112h; OT-3h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber

1.Complexidade assintótica,temporal e espacial

2.Tipos abstratos de dados e estruturas de dados fundamentais

3.Sistematizar a utilização das estruturas de dados na resolução de categorias de problemas reais

4.Técnicas fundamentais de desenho de algoritmos(divisão e conquista e função-memória)

5.Algoritmos de ordenação e suas condições de aplicação

Saber Fazer

6.Calcular a complexidade de algoritmos polinomiais ou exponenciais, iterativos ou recursivos

7.Modelar programas usando tipos abstratos de dados

8.Escolher,comparar,adaptar e utilizar estruturas de dados adequadas ao problema

9.Implementar estruturas de dados,em particular,recorrendo a gestão dinâmica de memória

10.Implementar algoritmos recursivos polinomiais.

Competências Complementares:

11.Efetuar escolhas fundamentadas

12.Concretização na implementação de um projeto

13.Avaliar soluções

14.Gestão do tempo

15.Comunicação escrita:relatório de projeto da disciplina

16.Comunicação oral:discussão do projeto da disciplina

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge

1. The basics of algorithm analysis.

2. Fundamental abstract data types and data structures.

3. How the main data structures are used to solve real-world problems.

4. The fundamental algorithm design techniques, namely divide-and-conquer and memoization.

5. Several sorting algorithms, their features and application conditions.

Application

6. To calculate the running time and the space requirements of a polynomial or exponential time algorithm.

7. To model programs using abstract data types.

8. To choose, compare, adapt and use suitable data structures for a given problem.

9. To implement data structures, in particular, using dynamic memory allocation.

10. To implement recursive polynomial time algorithms.

Soft-Skills

11. *To justify choices.*
12. *To do a programming project.*
13. *To evaluate solutions.*
14. *Time management.*
15. *Writing communication: course project report.*
16. *Oral communication: course project assessment.*

3.3.5. Conteúdos programáticos:

A. Introdução à Análise de Algoritmos

- *Complexidade temporal e espacial.*
- *Complexidade no melhor caso, no pior caso e no caso esperado.*

B. Introdução à Recursividade

- *Divisão e conquista.*
- *Função-memória.*

C. Tipos Abstratos de Dados

- *Pilha (disciplina LIFO).*
- *Fila (disciplina FIFO).*
- *Lista (acesso por posição).*
- *Dicionário (acesso por chave): não ordenado e ordenado.*
- *Fila com prioridade.*

D. Estruturas de Dados

- *Vetores circulares.*
- *Listas simplesmente e duplamente ligadas.*
- *Tabelas de dispersão.*
- *Árvores genéricas.*
- *Árvores binárias.*
- *Árvores binárias de pesquisa: árvores sem restrições; árvores AVL.*
- *Heaps.*

E. Algoritmos de Ordenação

- *Ingénuos: insertion sort; selection sort; bubble sort.*
- *Eficientes: heapsort; mergesort; quicksort.*
- *Por indexação: bucket sort; radix sort.*

3.3.5. Syllabus:

A. Algorithm Analysis

- *Running time and space requirements.*
- *Analysis in the best-case, in the worst-case, and in the average-case.*

B. Introduction to Recursion

- *Divide-and-conquer.*
- *Memoization.*

C. Abstract Data Types

- *Stack (LIFO).*
- *Queue (FIFO).*
- *List (access by position).*
- *Dictionary (access by key): unordered and ordered.*
- *Priority queue.*

D. Data Structures

- *Circular arrays.*
- *Singly and doubly linked lists.*
- *Hash tables.*
- *Generic trees.*
- *Binary trees.*
- *Binary search trees: trees without constraints; AVL trees.*
- *Heaps.*

E. Sorting Algorithms

- *Naïve algorithms: insertion sort; selection sort; bubble sort.*
- *Efficient algorithms: heapsort; mergesort; quicksort.*
- *Sorting without comparisons: bucket sort; radix sort.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os objetivos 1 e 6 são endereçados no capítulo A do Programa. Os objetivos 1, 6, 11 e 13 são ainda

endereçados ao longo de todo o semestre, seja na exposição dos alunos às estruturas de dados seja no desenvolvimento de exemplos e do próprio projeto, durante as aulas práticas.

Os objetivos 2 e 7 são endereçados nos capítulos C e D do Programa.

Os objetivos 2,3, 8, 9, 11 e 13 são endereçados no capítulo D do Programa, dado que cada estrutura de dados é apresentada com exemplos de implementação, de execução e discutindo as categorias de problemas em que se insere.

Os objetivos 4 e 10 são endereçados nos capítulos B (onde se discutem às várias técnicas de desenho de algoritmos consideradas) e D (onde se estuda a aplicação destas técnicas em estruturas de dados recursivas, tais como as árvores binárias de pesquisa) do Programa. Estes objetivos são também considerados no estudo dos algoritmos de ordenação recursivos (Capítulo E).

O objetivo 5 é endereçado no capítulo E do Programa.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Objetives 1 and 6 are addressed in chapter A of the syllabus. Objectives 1, 6, 11 and 13 are also addressed throughout the semester, resulting from the exposition of students to the data structures as well as, while developing examples and the course project, independently.

Objetives 2 and 7 are addressed in chapters C and D of the syllabus.

Objetives 2, 3, 8, 9, 11 and 13 are addressed in chapter D of the syllabus, as each data structure is presented with implementation and execution examples, and by discussing the categories of problems most suited to them.

Objetives 4 and 10 are addressed in chapters B (where the considered algorithm design techniques are discussed) and D (where the application of the techniques to recursive data structures, like binary search trees, is considered) of the syllabus. These objectives are also relevant in the study of recursive sorting algorithms (chapter E).

Objetive 5 is addressed in chapter E of the syllabus.

3.3.7. Metodologias de ensino (avaliação incluída):

O ensino consiste na exposição da matéria em aulas teóricas e na resolução de problemas em aulas práticas de laboratório. No laboratório, os alunos desenham, analisam e implementam algoritmos. Algumas aulas práticas são dedicadas ao acompanhamento do trabalho final.

O trabalho é obrigatório e realizado individualmente ou em grupos de 2 alunos, em 2 fases, entregues em datas marcadas e sujeitas a avaliação independente.

Os testes são sem consulta, escritos e individuais.

Avaliação:

- Notas combinadas das duas fases do trabalho, valendo cada uma 15% da nota final (NP).

- Notas combinadas de 2 testes, valendo o primeiro 30% da nota final e o segundo 40% da nota final (NT).

- Em época de recurso, a nota dos dois testes é substituída pela nota do exame, que vale 70% (NT) da nota final.

O aluno obtém aprovação se ambas as notas NP e NT forem superiores ou iguais a 9.5 valores em 20.

As melhorias de nota apenas se realizam por exame, sendo a nota final igual à nota do exame.

3.3.7. Teaching methodologies (including evaluation):

The course methodology consists of lectures and lab sessions. In the laboratory, students design, analyze and implement algorithms. Some lab sessions are dedicated project support.

The programming project is compulsory and can be developed individually or in groups of 2 students. The project is developed in two phases which are evaluated independently.

Written tests are solved individually.

Evaluation:

- Combined grades of the project phases. Each phase is worth 15% of the final grade (NP);

- Combined grades of the 2 individual tests. The first test is worth 30% of the final grade and the second test 40% of the final grade (NT).

- In the appeal season, the tests grade is replaced by the exam grade, which is worth 70% (NT) of the final grade.

A student is approved if both NP and NT are greater or equal to 9.5 points out of 20.

Grade improvement can only be achieved in the appeal exam, which is worth 100% of the final grade.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A metodologia de ensino envolve:

• A exposição teórica dos conteúdos programáticos, o que endereça, ao longo do semestre, os objetivos 1 a 5.

São ainda relevantes, neste contexto, os objetivos 6 a 10, uma vez que toda a matéria é exposta com exemplos e considerando categorias de problemas. As aulas teóricas contribuem ainda para os objetivos 11 e 13 uma vez que as escolhas feitas na exposição dos exemplos são fundamentadas tendo como base a avaliação das soluções.

- O desenvolvimento de exercícios aplicados em aulas práticas, que vão treinar os conhecimentos adquiridos na componente teórica (relativa aos objetivos 1-4). O treino é desenvolvido no contexto dos objetivos “Fazer” (6 a 10), ou seja, conceber tipos abstratos de dados dimensionados para um dado problema, e escolher estruturas de dados, adaptando-as para a implementação do mesmo. O cálculo da complexidade serve de suporte às escolhas feitas. Este tipo de atividade envolve a aquisição das soft-skills 11 e 13.
- O desenvolvimento de um trabalho prático, nas aulas práticas e autonomamente. Também aqui a componente teórica serve de suporte (objetivos 1 a 4), assim como as atividades de treino desenvolvidas nas aulas práticas (e os objetivos endereçados nesse contexto). O trabalho é desenvolvido em duas fases, com datas de entregas e avaliações independentes. O trabalho é entregue de forma automática, num sistema de gestão de concursos de programação. Só os trabalhos aceites por este sistema são considerados para avaliação. As entregas de trabalhos, efetuadas desta forma, implicam o desenvolvimento de capacidades de concretização e de gestão de tempo (objetivos 12 e 14), de forma a cumprir, não só os prazos temporais do trabalho, mas também os requisitos mínimos determinados pelo uso do sistema de gestão de concursos. Finalmente, os alunos são ainda avaliados em termos da comunicação escrita, pela elaboração de um relatório de projeto, e oral, porque têm de defender os conhecimentos adquiridos durante o desenvolvimento do projeto, numa discussão do mesmo (objetivos 15 e 16).
- A resposta a perguntas de teste ou exame. Nesta componente de avaliação são avaliados grande parte objetivos já mencionados atrás, seja a componente de exposição nas aulas teóricas (objetivos 1 a 5), quer a componente de Fazer (objetivos 6 a 10). As perguntas que envolvem a escolha de estruturas de dados para a resolução de um dado problema endereçam os objetivos 11 e 13. As perguntas de desenvolvimento de código, são dirigidas à extensão de estruturas conhecidas, ou ao desenvolvimento de funcionalidades específicas relacionadas com um problema. Esta componente de avaliação é escrita, individual, e respondida dentro do tempo atribuído para a resposta às perguntas, o que implica a gestão do tempo e a capacidade de comunicação escrita (objetivos 14 e 15).

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit’s intended learning outcomes:

The course methodology involves:

- *Theoretical presentation of the syllabus, which addresses, throughout the semester, objectives 1 to 5. In this context, objectives 6 to 10 are also relevant, as the whole syllabus is exposed using examples and considering categories of problems. Lectures also contribute to objectives 11 and 13, as the choices made in examples are justified.*
- *The development of practical examples in lab sessions, practicing the knowledge acquired in lectures (associated with objectives 1-4). Training is developed in the context of objectives 6 to 10, by conceiving abstract data types for a given problem and choosing and adapting data structures for the implementation of a solution. Complexity analysis will support choices. This type of activity involves soft-skills 11 and 13.*
- *The development of the course project, in labs and autonomously. The knowledge acquired during lectures is still important in this context (objectives 1 to 4) as well as all the training activities developed in labs (and the objectives addressed in that context). The project is submitted automatically through a Programming contest management system. Only the projects accepted by this system are considered for evaluation. This type of submission leads to the development of implementation and time management skills (objectives 12 and 14), as students need, not only to meet the deadlines of the project, but also the minimal requirements set by the contest management system. Finally, students are still evaluated in written and oral communication (objectives 15 and 16) through the submission of project report and by the oral justification of the knowledge acquired during the development of the project.*
- *Answering exam or test questions. In this evaluation component, most of the objectives already mentioned are evaluated, meaning the knowledge of the syllabus (objectives 1 to 5), and the Application component (objectives 6 to 10). The questions regarding the choice of data structures for the solution of a problem address objectives 11 and 13. Programming questions aim at the extension of known data structures or at the development of problem related specific functionalities. This component is written, individual e time restrained, which implies time management and written communication skills (objectives 14 and 15).*

3.3.9. Bibliografia principal:

Michael T. Goodrich and Roberto Tamassia. Data Structures and Algorithms in Java (5th edition). John Wiley & Sons, 2010.

Mark Allen Weiss. Data Structures and Algorithm Analysis in Java (2nd edition). Addison-Wesley, 2007.

Mark Allen Weiss. Data Structures and Problem Solving Using Java (4th edition). Addison-Wesley, 2010.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms (3rd edition). The MIT Press, 2009.

Mapa IV - Fundamentos de Sistemas de Operação / Operating Systems Foundations

3.3.1. Unidade curricular:

Fundamentos de Sistemas de Operação / Operating Systems Foundations

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Alberto Cardoso e Cunha (Responsável e Regente) (T-84h; OT-2h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Vítor Manuel Alves Duarte (P-112h; OT-2h)

Paulo Orlando Afonso Lopes (P-84h; OT-2h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *as funcionalidades dos sistemas de operação (SO), suas interfaces de utilização e programação;*
- *as abstrações do SO para a execução de programas e gestão de recursos;*
- *os princípios da programação concorrente e da coordenação de processos;*
- *os princípios de conceção e de organização interna de um sistema de operação.*

Saber fazer:

- *Relacionar os aspetos teóricos e os aspetos práticos e melhorar a capacidade de realização de trabalhos práticos laboratoriais;*
- *Utilizar o ambiente e as ferramentas de apoio ao desenvolvimento e gestão da execução de programas, ao nível das interfaces do sistema de operação, e com os modelos de programação ao nível das chamadas ao sistema de operação, envolvendo concorrência, comunicação e sincronização entre processos.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *the operating system functionalities, and the user and programming (system call) interfaces;*
- *the operating system abstractions for program execution and resource management;*
- *the principles of concurrent programming and process coordination;*
- *the essentials of the operating system strategies for memory and process management.*

Application:

- *relate theoretical and practical aspects both conceptually and experimentally through laboratorial practice;*
- *use the operating system environment and tools for program development and execution, and programming at the system call level with process concurrency, communication and synchronization;*

3.3.5. Conteúdos programáticos:

- 1. Introdução aos sistemas de operação (SO): tipos de SO, núcleo e serviços.*
 - 2. Processos e threads; multiprogramação, escalonamento e comutação de contextos.*
 - 3. Princípios da programação concorrente: traços de execução e atomicidade; coordenação: exclusão mútua, produtores e consumidores, leitores e escritores, clientes e servidores; sincronização: regiões críticas, condições, espera ativa, bloqueio, locks, semáforos e monitores; comunicação: memória partilhada, pipes e mensagens em SO centralizados.*
 - 4. Ficheiros: organização e operações do sistema de ficheiros; memória secundária e terciária: características dos discos físicos e dispositivos amovíveis; princípios da gestão das entradas e saídas..*
 - 5. Princípios das estratégias de gestão da memória e de processos: memória virtual; algoritmos de escalonamento; regulação da carga de multiprogramação.*
- Laboratório: Programação ao nível das chamadas ao SO: ficheiros, composição e comunicação entre processos/threads.*

3.3.5. Syllabus:

1. *Introduction to operating systems (OS): types, organization and services.*
2. *Processes and threads: multiprogramming, scheduling and context switching.*
3. *Principles of concurrent programming: execution traces and atomicity; coordination problems: mutual exclusion, producers and consumers, readers and writers, clients and servers; synchronization: critical regions, conditions, busy waiting, blocking, locks, semaphores and monitors; communication: shared memory, pipes, messages in centralized OS.*
4. *Files: file system organization and primitives; characteristics of mass storage, disks and removable devices; essentials of the input/output management.*
5. *Principles of memory and process management: virtual memory, process scheduling and multiprogramming. OS programming lab: program with files and processes/threads, communication/synchronization mechanisms at the system level.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Globalmente os capítulos do programa contribuem para cumprir os objetivos enunciados. Apresentam-se os conceitos fundamentais dos sistemas de operação (SO). O capítulo inicial enquadra os temas e o final dá uma visão integrada das estratégias do núcleo do SO. Os capítulos 2, 3 e 4 desenvolvem os conceitos de processo e de ficheiro, a nível das abstrações de programação e dos princípios da sua realização pelo SO. Consideram-se, por um lado, as interfaces de programação das chamadas ao sistema, face aos conceitos de processo e de ficheiro (capítulos 2 e 4) e a coordenação de processos concorrentes (capítulo 3) e, por outro lado, o SO, como um gestor de recursos, e os seus mecanismos e estratégias internos. A prática laboratorial reforça o cumprimento dos objetivos, pela ligação da teoria à prática. Exercita o desenvolvimento de programas concorrentes baseados nas interfaces de programação de processos, threads e ficheiros e a capacidade de avaliação experimental dos resultados.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Overall the chapters in the syllabus fulfill the course objectives in presenting the essentials of operating systems (OS). The initial chapter gives an overview and the final chapter gives an introductory integrated view of the internal OS strategies. Chapters 2, 3, and 4 discuss the process and file abstractions and how they are supported by the OS. On one hand, the process and file programming abstractions at the system call level (chap 2 and 3), and concurrent process coordination (chap 3) are studied, and on the other hand the OS resource management strategies and mechanisms are introduced. The laboratory part contributes to consolidate the above objectives by linking theory and practice, and exercising the ability for development and experimental evaluation of concurrent programs based on the system level interfaces for processes, threads and files,

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é desenvolvido através de sessões teóricas, sessões práticas laboratoriais para a resolução de problemas, estudo autónomo do aluno, trabalhos em grupo e sessões de contacto individual entre docentes e alunos. Os conceitos teóricos são apresentados e discutidos em sessões teóricas, que permitem a apresentação e enquadramento dos temas tratados, concentrando a atenção do aluno nos conceitos fundamentais, na sua aplicação e utilidade, bem como a forma como são concretizados a nível do sistema de operação. Os conceitos teóricos são ilustrados através da realização de um conjunto de trabalhos práticos, de natureza laboratorial, nos quais os alunos aplicam os conceitos teóricos para resolver os problemas propostos, sobre o ambiente laboratorial, exercitando os temas do programa. A avaliação assenta em testes presenciais individuais e em trabalhos laboratoriais ao longo do semestre, em grupos, havendo um relatório escrito por cada trabalho e sua discussão oral.

3.3.7. Teaching methodologies (including evaluation):

The course encompasses lectures and practical lab sessions, and involves the students in autonomous individual work, in group work, and in individual contact sessions with the instructors. Theoretical concepts, as presented and discussed in the lectures, contribute to the understanding of the course matters by guiding the students through the essential concepts, their application, and how they are supported at the operating system level. Such themes are then exercised in practice through small programming projects using the lab operating system environment. Student evaluation relies on two components. One is based on individually written tests, and the other one is based on laboratory assignments performed by groups of students along the course duration. The latter component requires both a written report and an oral presentation per assignment.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Para os temas dos diversos capítulos, as sessões teóricas procuram expor e salientar as questões centrais e as dimensões que caracterizam as possíveis soluções, com o objetivo, por um lado, de contribuir para a melhor compreensão dos conceitos e, por outro lado, visando estimular a autonomia dos alunos para a sua

concretização nas instâncias específicas dos problemas a resolver nos trabalhos laboratoriais. Nas sessões teóricas e na sequência destas, o aluno é incentivado a questionar os conceitos e as técnicas apresentados. A mesma metodologia é aplicada para cada capítulo do programa. O aluno, para cada trabalho, vê-se confrontando com a necessidade de aplicar o conhecimento dos conceitos e o quadro das soluções gerais, que foram apresentadas/discutidas nas sessões teóricas, para o caso de cada problema concreto a resolver. Suporta-se uma articulação apertada entre os conceitos teóricos e as suas realizações práticas laboratoriais. Esta articulação é evidenciada pela correspondência que se estabelece entre os trabalhos práticos e os respetivos capítulos do programa, tanto a nível das matérias, como a nível da sequência temporal de desenvolvimento das sessões teóricas e práticas. A dependência da nota final das duas componentes apresentadas (testes e trabalhos), representa a importância que se atribui à articulação da teoria com a prática, do trabalho individual com o coletivo e das capacidades de comunicação escrita (testes o relatório) e oral (apresentação de trabalhos).

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In the presentation of the main themes, the course lectures enable a first-level understanding of the concepts and their application, and on the other hand aim to stimulate the student towards getting a deeper understanding and a critical perspective by discussing the concepts and alternative solutions for their implementation in specific problem scenarios. The latter are further explored in the practical labs illustrating the trade-offs between theory and practice, which is also achieved by enforcing a tight relationship between the lectures development and the practical assignments, along the course duration.

The two evaluation components (individual tests and lab assignments) emphasize a balance between theory and practice, individual and collective work, and abilities for written and oral presentations.

3.3.9. Bibliografia principal:

Operating System Concepts - Essentials, A. Silberschatz, P. B. Galvin, G. Gagne, John Wiley & Sons, 2010.

UNIX System Programming (2nd Edition), K. Haviland, D. Gray, B. Salama, Addison-Wesley, 1999.

UNIX Systems Programming (2nd Edition), K. A. Robbins, S. Robbins, Prentice-Hall/Pearson Education, 2003.

Mapa IV - Lógica Computacional / Computational Logic

3.3.1. Unidade curricular:

Lógica Computacional / Computational Logic

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Manuel Corrêa Calvente de Barahona (Responsável e Regente)(T-56h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Susana Maria dos Santos Nascimento Martins de Almeida (P-84h)

Ludwig Krippahl (P-84h)

Nuno Miguel Cavalheiro Marques (P-126h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- Sintaxe e a semântica da lógica proposicional e de primeira ordem*
- Método de resolução para a lógica proposicional e de primeira ordem*
- Sistemas de dedução natural da lógica proposicional e de primeira ordem*

Saber Fazer:

- Escrever fórmulas lógicas a partir de descrições em língua natural*
- Calcular semântica, axiomática, e sintaticamente a validade lógica de fórmulas*
- Usar resolução para estabelecer a validade lógica de fórmulas*

Competências Complementares:

- Capacidade de raciocínio abstrato e rigoroso*
- Capacidade de manipulação de estruturas formais*
- Aprender a aprender*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge

- *Syntax and semantics of propositional and first-order logic*
- *Resolution methods for propositional and first-order logic*
- *Natural deduction systems for propositional and first-order logic*

Application:

- *Specification logical formulae from descriptions in natural language*
- *Assessment of the logical validity of formulae, semantically, axiomatically, and syntactically*
- *Use resolution algorithms to establish the logical validity of formulae*

Soft-skills:

- *Ability of abstract and rigorous reasoning*
- *Ability of manipulating formal structures*
- *Learn to learn*

3.3.5. Conteúdos programáticos:

1. Lógica Proposicional

1.1. Sintaxe:

- *Definição indutiva de linguagem proposicional*
- *Termos e prova da sua pertença a linguagem*
- *Provas por indução estrutural*

1.2. Semântica:

- *Tabelas de verdade e álgebra de Boole*
- *Valoração e estrutura de interpretação: satisfação*
- *Validade e consequência lógica; equivalência*
- *Leis axiomáticas: distribuição de conectivos e sua correção*

1.3. Algoritmos de decisão e Sistemas de Prova

- *Dedução natural: Regras de introdução e eliminação*
- *Resolução: Forma clausal, Algoritmos de Horn*

2. Lógica de primeira ordem

2.1 Sintaxe:

- *Termos e Linguagem de 1ª ordem*
- *Variáveis livres e substituição*

2.2 Semântica:

- *Valoração e estrutura de interpretação: satisfação*
- *Validade e consequência lógica; equivalência*
- *Leis axiomáticas: distribuição de conectivos e sua correção*

2.3 Algoritmos de decisão e Sistemas de Prova:

- *Resolução: Skolemização, Unificação*
- *Dedução natural: Regras de introdução e eliminação*

3.3.5. Syllabus:

1. Propositional Logic

1.1 Syntax:

- *Inductive definition of propositional language*
- *Terms and proof of being in a language*
- *Proofs by structural induction*

1.2 Semantics:

- *Truth tables and Boolean algebra*
- *Valoration and interpretation structure: satisfaction*
- *Validity and logical consequence; equivalence*
- *Axiomatic laws: distribution laws and soundness proofs*

1.3 Deductive systems and Decision Algorithms

- *Natural deduction: Introduction and elimination rules*
- *Resolution: Clausal form, Horn algorithm*

2. First Order Logic

2.1 Syntax:

- *Alphabet and first order language*
- *Terms from natural language descriptions*
- *Free variables and substitution*

2.2 Semantics:

- *Valoration and interpretation structure: satisfaction relation*
- *Validity and logical consequence; equivalence*
- *Axiomatic laws: presentation of the distribution laws and soundness proofs*

2.3 Deductive systems and Decision Algorithms

- *Natural deduction: Introduction and elimination rules*

•Resolution: Clausal form, Skolemisation, Unification

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Esta UC cobre os requisitos em lógica matemática previstos no ACM 2008 CS Curriculum na área de Estruturas Discretas (DS/BasicLogic [core]). Em complemento dos acima referidos, este documento inclui como objetivos: (a) aplicação de métodos formais das lógicas proposicional e de predicados; (b) modelação de contextos, incluindo computacionais, em que a lógica pode ser aplicada (e.g. correção de programas, bases de dados e algoritmos); (c) aplicação da lógica para resolução de problemas (puzzles); (d) limitações da lógica de predicados.

A matéria lecionada satisfaz estes objetivos nomeadamente através do estudo da sintaxe e semântica das lógicas proposicional e de predicados, bem como da formalização de inferência através da consequência lógica (a); a exemplificação e resolução destes conceitos em vários exemplos (b) e (c); e a formalização de especificações em língua natural, sendo analisadas as limitações de representação e raciocínio nas lógicas referidas.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The unit addresses the mandatory requisites in mathematical logical foreseen in the ACM 2008 CS Curriculum (Discrete Structures: DS/BasicLogic [core]). In addition to those above, this document includes as objectives: (a) Application of formal methods of propositional and predicate logic; (b) Use of formal tools of symbolic logic to model real-life situations, including computing contexts (program correctness, database queries, and algorithms); (c) Use of formal logic to solve problems (e.g. puzzles); and (d) discussion of the importance and limitations of predicate logic.

The syllabus satisfy these objectives, namely through the study of the syntax and semantics of propositional and predicate logic, and the formalisation of inference as logic consequence (a); The application of logic in several problems, including their solving (b)/(c); and the formalisation of specifications in natural language, overviewing the representational and reasoning limitations of these logics (d).

3.3.7. Metodologias de ensino (avaliação incluída):

Os conteúdos (motivações, definições, resultados e exemplos) são apresentados nas aulas teóricas – duas de uma hora por semana, para permitir que os alunos assimilem a matéria. Por ser essencial o treino dos conceitos para a sua consolidação, são realizadas fichas de exercícios em duas aulas práticas por semana. Muitos exercícios são resolvidos usando as várias abordagens apresentadas, para ajudar a clarificar os conceitos e relacionar e contrastar essas abordagens.

A avaliação é feita através de 4 testes (ou exame de recurso). O 1º e o 3º testes (de estruturas muito semelhantes) cobrem a sintaxe e a semântica da das lógicas proposicional e de predicados, respetivamente. Analogamente, os 2º e o 4º testes cobrem os algoritmos e o sistema dedutivo das lógicas proposicional e de predicados.

Por ser uma cadeira formal e de fundamentos, tanto o trabalho dos alunos como a avaliação seguem as metodologias típicas do ensino das ciências básicas.

3.3.7. Teaching methodologies (including evaluation):

The different topics (motivation, definitions, results and examples) are presented in two weekly theoretical classes of 1-hour duration, to allow students to absorb the contents). Training and consolidation of concepts is in two weekly laboratory classes (1.5 hours each), by means of selected exercises. Many of these are solved with the different approaches taught, to help students to clarify the concepts and compare the approaches. The assessment is done through 4 mi-term tests (or a final exam). The 1st and 3rd, of similar structure, cover the syntax and semantics of propositional and predicate logic, respectively. Analogously, the 2nd and 4th, address the algorithms and deductive systems of propositional and predicate logics.

Being a foundational course on formal subjects, both the work of the students and their assessment follow the methodologies typical of basic sciences teaching.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A aprendizagem do rigor e das abordagens formais requer a exposição à apresentação precisa dos conceitos e às provas matemáticas dos resultados. Os alunos devem não esforçar-se por entender os conceitos como sobretudo exercitá-los autonomamente, para ficarem capazes de usar e aplicar os conceitos nos contextos em que tenha necessidade.

A UC pretende ser uma introdução à Lógica Matemática e aos métodos computacionais para avaliar e decidir a validade de afirmações ou raciocínios. A Lógica de Predicados (de Primeira Ordem) é uma ferramenta de especificação central quer na Matemática quer na Informática, sendo essencial tanto para o desenvolvimento de software como de hardware. Para que a aprendizagem seja gradual, começa-se por um subconjunto

“clássico” da lógica de predicados, a Lógica Proposicional.

Para ser claro como se escrevem fórmulas, usam-se definições indutivas, que permitem também não só verificar se dada expressão é uma fórmula recorrendo a uma árvore de dedução, como permitem que muitas noções auxiliares úteis sejam definidas como funções recursivas e permitem ainda fazer as provas das propriedades apresentadas por indução estrutural.

Para ser simples e não haver ambiguidade na interpretação de uma fórmula, define-se a sua semântica por interpretação numa álgebra de Boole, permitindo avaliar fórmulas por simples cálculo algébrico. Define-se a noção de consequência semântica como formalização de raciocínios de inferência, e relaciona-se a noção com o conectivo de implicação. Porque há fórmulas diferentes que representam o mesmo valor de verdade, define-se uma noção de equivalência lógica e mostram-se várias leis que capturam interações entre os operadores. Porque na informática nos interessa a avaliação automática de fórmulas, mostram-se dois algoritmos de decisão, um muito eficiente mas não universal (o de Horn), e outro universal e que está na base da Programação em Lógica (a resolução).

Finalmente, apresenta-se um método de prova da validade de fórmulas e de raciocínios puramente sintático (i.e., dispensando a avaliação): a dedução natural.

Os alunos ficam assim com uma panorâmica global típica da Lógica como ferramenta de especificação e verificação: sintaxe, semântica, algoritmos de decisão e sistema dedutivo. Este conhecimento é essencial para o desenvolvimento e análise de algoritmos, para a representação de conhecimento, e para o desenho e verificação de circuitos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit’s intended learning outcomes:

Learning formal subjects and requires a precise presentation of the concepts and the mathematical proof of results. The students should endeavour to understand the concepts but also to train and exercise them autonomously, so as to be able to use and apply these concepts in contexts they will be exposed to in the future. The course aims at being an introduction to Mathematical Logic and to the computational methods to validate statements and reasoning. Predicate Logic is a key tool for specification, both in Mathematics and Computer Science, namely for the development of software and hardware. To achieve an incremental learning, the course starts with Propositional Logic, a classical subset of Predicate Logic.

To support the identification of valid formulae, inductive definitions are used to allow (by means of deduction trees) not only verifying whether an expression is a valid formula, but also to define many useful concepts as recursive functions, and enable the proving of properties by means of structural induction.

To simplify formulae interpretation, and allow no ambiguity in his process, the semantics is defined by means of a Boolean algebra, which allows the evaluation of formulae through simple algebraic calculus. The notion of semantical inference is defined as the formalisation of inference reasoning, and is related to the implication connective. To compare different formulae with the same truth value, the notion of logic equivalence is presented, together with several laws that capture operator interactions.

For the automated evaluation of formulae, of great relevance in Computer Science, two decision algorithms are addressed: one (Horn) that is very efficient but not universal, and another, Resolution, that is the basis of Logic Programming. Finally, another proof method, purely syntactical (requiring no evaluation), natural deduction, is presented to verify the validity of formulae and reasoning.

The students thus achieve a global view of Logic as a tool for specification and verification: syntax, semantics decision algorithms and deductive system. This knowledge is paramount to the development and analysis of algorithms, for knowledge representation and problem solving, namely design and verification of digital circuits.

3.3.9. Bibliografia principal:

- *Language Proof and Logic (4th edition). Jon Barwise and John Etchemendy. CSLI Publications, 2003.*
- *Mathematical Logic: a course with exercises. Part I: propositional calculus, Boolean algebras, predicate calculus. René Cori e Daniel Lascar. Oxford Press, 2007.*
- *A First Course in Logic: An Introduction to Model Theory, Proof Theory, Computability, and Complexity. Shawn Hedman. Oxford Texts in Logic, 2004.*
- *Logic in Computer Science: modelling and reasoning about systems (2nd edition). Michael Huth and Mark Ryan. Cambridge University Press, 2004.*
- *Lógica Computacional: Proposicional e de 1ª ordem. Apontamentos de Paula Gouveia e F. Miguel Dionísio. Instituto Superior Técnico.*

Mapa IV - Física / Physics

3.3.1. Unidade curricular:

Física / Physics

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:
Jorge Alexandre Monteiro de Carvalho e Silva (T-84h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:
André João Maurício Leitão do Valle Wemans (P-84h)
José Luís Constantino Ferreira (P-126h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

A unidade curricular tem por objectivo proporcionar aos estudantes a aquisição dos seguintes conhecimentos, aptidões e competências:

- conhecer as forças de contacto e à distancia relevantes para o estudo do movimento do ponto material, de sistemas de partículas e do corpo rígido e ser capaz de as identificar em situações realistas*
- conhecer e aplicar as leis de Newton do movimento no que diz respeito à dinâmica do ponto material, dos sistemas de partículas e do corpo rígido*
- conhecer e aplicar à resolução de problemas as leis e princípios de conservação, nomeadamente da energia mecânica, momento linear e momento angular;*
- a modelação computacional de sistemas físicos usando a plataforma Modellus, em particular a resolução numérica das equações do movimento;*
- a aprendizagem de métodos experimentais em Física;*
- a sistematização da recolha e tratamento de resultados experimentais, em particular o cálculo da incerteza associada a uma medida física;*
- a elaboração de relatórios científicos.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This curricular unit aims at providing students with the following knowledge, skills and competencies:

- know the forces that act through contact and at the distance that are relevant for the study of the movement of a particle, a system of particles and the rigid body and being able to identify them in realistic situations;*
- know and apply Newton's laws of movement to the dynamics of the movement of a particle, a system of particles and the rigid body*
- know and apply to the resolution of problems conservation laws and principles, like those of mechanical energy, linear and angular momentum;*
- computational modelling using The Modellus platform, in particular the numerical resolution of equations of motion;*
- learning experimental methods in Physics;*
- the acquisition and treatment of experimental data, in particular the evaluation and expression of uncertainty of measurement of a physical quantity;*
- the writing of scientific reports.*

3.3.5. Conteúdos programáticos:

- 1. Conceitos Gerais: Grandezas, medidas, incertezas, dimensões e unidades. O espaço e o tempo. Referenciais. Os modelos da mecânica.*
- 2. Cinemática: Os conceitos de velocidade e aceleração. Determinação da velocidade e do vector posicional a partir da aceleração. Movimento relativo de translação uniforme, transformação de Galileu.*
- 3. Dinâmica da Partícula: Referenciais de inércia. Conservação do momento linear, do momento angular e da energia. Referenciais não inerciais.*
- 4. Dinâmica de Sistemas de Partículas: Centro de massa e referencial do centro de massa. Conservação do momento linear, do momento angular e da energia. Colisões. Sistemas de massa variável.*
- 5. Dinâmica do corpo rígido: Conservação do momento linear, do momento angular e da energia na translação e rotação em torno de um eixo. Movimento geral; de rolamento. Equilíbrio do corpo rígido.*
- 6. Movimentos Oscilatórios: Movimento oscilatório harmónico.*

3.3.5. Syllabus:

- 1.General concepts:Measurement,units,uncertainties,dimensions.Space and Time.Reference frames.Models of Mechanics.*
- 2.Kinematics:Velocity and acceleration in general motion.Velocity and position calculated from the acceleration.Relative motion,Galileu's transformation.*
- 3.Dynamics of the particle:Inertial reference frames.Principle of conservation of linear momentum.Fundamental forces; contact forces;friction;drag force and terminal speed.Non-inertial reference frames.Conservation of angular momentum.Work and energy;potential energy and force;conservation of energy.*

4. Systems of particles: Center of mass and center of mass reference frame. Conservation of linear momentum, angular momentum and energy. Two particle systems; reduced mass. Collisions. Systems with varying mass.

5. Dynamics of the rigid body: Linear momentum, angular momentum, work and energy in translation and rotation. Rotational inertial momentum. General movement; rolling. Equilibrium.

6. Oscillatory Motion: Simple harmonic motion.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A Física, tal como a Matemática, está na base de todas as áreas científicas. Por outro lado, a Física, usando o método e raciocínio científicos, tem como base a observação e a medida de grandezas associadas aos vários fenómenos. Qualquer unidade curricular de Física deve conseguir transmitir essas ideias e é possível fazê-lo com qualquer tipo de matéria. Faz sentido, numa primeira unidade curricular, abordar os princípios e conceitos mais gerais da Física, porque sustentam as várias subáreas da Física e porque necessariamente aparecerão em várias outras áreas científicas. É esse o papel da Mecânica. Por outro lado, de modo a que os conceitos e métodos teóricos e experimentais sejam apreendidos, a matéria não pode ser muito vasta e a unidade curricular terá de ter uma componente experimental. Embora em primeira aproximação pudesse parecer mais interessante ensinar a Física da perspectiva da Física Moderna, isto é incluindo a Relatividade e a Mecânica Quântica, passar para além do nível da divulgação científica envolve uma preparação de Física clássica (mais intuitiva) e mais preparação de Matemática do que os alunos têm após um ano de um curso universitário. Assim, para cumprir os objectivos informativos (conhecimento) e formativos (competências transversais) que se pretende, a matéria de Mecânica Clássica parece ser a mais adequada.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Physics, along with Mathematics, is the cornerstone of all scientific areas. Also, Physics, employing the scientific method and reasoning, is based on observation and measurement of quantities related to the observed phenomena. Any curricular unit of Physics has to convey those ideas to the students and that may be done with any Physics subjects. In the first curricular unit of Physics, it makes sense to lecture about the most general concepts and principles of Physics as they give support to other Physics subjects and because they will appear necessarily in other scientific areas. That is the role of Mechanics. Besides, in order that concepts and experimental and theoretical methods may be really understood, the subjects to be covered have to be limited and the curricular unit must have a laboratorial component. Although in first approximation, it might appear more interesting to teach Physics from the perspective of Modern Physics, including Relativity and Quantum Mechanics, in order to go beyond the level of general public scientific lectures, the students should have a pre-education in Classical Physics (more intuitive) and Mathematics of a level superior to what they have after their first university year. In conclusion, to fulfill the informative (knowledge) objectives and the formative (transversal competencies) ones, Classical Mechanics seems to be the most reasonable subject.

3.3.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas desenrolam-se de acordo com o método de ensino conhecido como “aprendizagem em equipa” (www.teambasedlearning.org) que é uma implementação do princípio da aprendizagem activa em que o aluno participa nas diversas actividades desenvolvidas na aula. Antes da aula o aluno prepara-se lendo os capítulos do livro indicados pelo professor e realizando um pequeno teste de certificação. Durante a aula realizam-se diversas actividades, em grupos de 5 a 7 elementos:

- repetição, em grupo, do teste individual;
- uma experiência, simulação ou visualização de um filme servem de base à realização de outras actividades;
- modelação computacional usando a plataforma Modellus
- breves apresentações (pelo professor) de temas seleccionados.

A avaliação centra-se em 4 componentes:

- os relatórios das actividades laboratoriais;
- os testes individuais de preparação
- as actividades realizadas em grupo durante as aulas teóricas
- 2 testes formais ou exame de recurso.

3.3.7. Teaching methodologies (including evaluation):

The classes will evolve according to the teaching method known as Team based learning (www.teambasedlearning.org) which is an implementation of the principle of active learning in which the students participate actively in the activities taking place during class. Before class, the students has to read the book chapters indicated by the teacher and complete an assessment test. During class, several activities (performed by the students in groups with 5 up to 7 members) will take place:

- repetition of the individual assessment test;
- experiments, simulations, movie visualizations will be used to develop further activities

- *computational modelling with the Modellus platform;*
- *brief lectures by the teacher of select subjects.*

Student evaluation will be based on:

- *reports of experimental activities*
- *the individual assessments tests*
- *the group activities;*
- *two formal tests or a final exam.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os estudantes têm dificuldades em aprender somente a partir do que ouvem nas aulas e têm especial dificuldade com a Física e a Matemática. Assim, para que apreendam e compreendam os conceitos e metodologias é importante o envolvimento activo dos estudantes no processo de ensino-aprendizagem. O funcionamento das aulas teóricas implementa o princípio da aprendizagem activa, em concreto o método da aprendizagem em equipa, para cumprir este propósito. As aulas de modelação computacional usam ferramentas bem conhecidas dos estudantes (a programação) para estudar sistemas físicos de forma dinâmica, que lhes permite observar o desenrolar dos movimentos e das interações, usando abordagens analíticas e numéricas. As aulas laboratoriais, contribuindo para a clarificação dos conceitos, trazem também aos estudantes as ideias por detrás da medida das grandezas físicas: as limitações experimentais, as aproximações simplificadoras, as incertezas.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Students have difficulty in learning solely from what they hear in class and have particular difficulties with Physics and Mathematics. Therefore, in order for them to apprehend and understand the concepts and methodologies, their active participation in the teaching-learning process is paramount. Classes are therefore structured according to the principle of active learning, namely the team based learning method, in order to fulfil this goal. Computational modelling classes uses programming, a tool these students know well, to study and solve physics problems in a dynamic environment that allows them to follow the movement and interactions of objects, using both analytical and numerical methods.

Laboratory classes contribute to clarify some concepts, bringing also to the student the ideas behind the measurement of a physical quantity; the experimental limitations, the simplifying approximations, the uncertainties.

3.3.9. Bibliografia principal:

- *"University Physics", Hugh Young e Roger Freedman. Pearson/Addison Wesley*
- *"Physics for Scientists and Engineers", Paul A. Tipler. W.H. Freeman and Company*
- *"Fundamentals of Physics", Halliday, Resnick e Walker. Wiley*

Mapa IV - Probabilidades e Estatística E / Probability and Statistics E

3.3.1. Unidade curricular:

Probabilidades e Estatística E / Probability and Statistics E

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Carlos Manuel Agra Coelho (apenas responsável, não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Pedro José dos Santos Palhinhas Mota (Regente) (TP – 112h)

Dora Susana Raposo Prata Gomes (TP – 112h)

Isabel Cristina Maciel Natário (TP – 112h)

Pedro Alexandre da Rosa Corte Real (TP – 112h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

O objetivo principal da disciplina é a transmissão de conceitos básicos de Probabilidades e Estatística, necessários a um aprofundamento posterior do conhecimento nesta área. Pretende-se, que os alunos adquiram competências básicas que permitam entender e analisar estas e outras técnicas estatísticas necessárias à sua atividade profissional, nomeadamente, a utilização dos métodos estatísticos de recolha,

análise e interpretação de dados.

Numa primeira fase do tratamento estatístico, pretende-se que os alunos consigam formalizar corretamente problemas que envolvam o resultado de experiências aleatórias, utilizando para isso o conhecimento adquirido com a teoria das probabilidades.

Numa segunda fase, introduz-se um conjunto de técnicas estatísticas que permitem o estudo de parâmetros numa população (Inferência estatística). Entre estas técnicas, destaca-se a regressão linear, que constitui um primeiro exemplo de modelação do real através da Estatística.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The objective of this subject is to teach the basics about the statistics and probability. The students will be prepared to easily handle the requirements of a professional activity, concerning probabilities and statistics. Regarding Probabilities it's intended that students develop capacities to formulate problems concerning the results of random experiences.

Pupils should be able as well to manipulate statistical techniques, in order to analyze parameters of a population. For instance be able to use linear regression as a first approach to modulation of real data throughout statistics.

3.3.5. Conteúdos programáticos:

Introdução à teoria das probabilidades, Variáveis aleatórias e suas distribuições de probabilidade, Momentos de variáveis aleatórias, Algumas distribuições importantes, Vetores aleatórios, Teorema Limite Central, Geração de números pseudoaleatórios, Estimação pontual, Estimação por intervalo de confiança, Testes de hipóteses, Regressão linear simples.

3.3.5. Syllabus:

Basic notions of probability, Random variables and their probability distributions, Moments of random variables, Some important distributions, Random vectors, Central limit theorem, Random number generation, Point estimation, Interval estimation, Hypothesis testing, Simple linear regression.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os objetivos definidos são os clássicos num curso introdutório em Probabilidades e Estatística para os cursos de Engenharia. Foram afinados a partir de sugestões dadas pelos docentes das áreas a que o curso se destina. Os tópicos 2, 3, 4 e 5 sistematizam as noções essenciais da teoria das probabilidades subjacentes aos modelos estáticos utilizados na prática. No tópico 1 apresenta-se a linguagem e as noções básicas das probabilidades requeridas para contextualizar o estudo a desenvolver. O tópico 6 é um dos pilares da estatística moderna ao justificar a prevalência do modelo normal mas, sobretudo, ao permitir a construção de intervalos de confiança e, por dualidade, a realização dos testes de hipóteses nas situações mais relevantes na prática. A geração de números pseudoaleatórios, abordado no tópico 7, faz parte das ferramentas básicas da engenharia informática moderna. A fundamentação e o estudo de um significativo número de exemplos de estimação e de testes de hipóteses, executada nos tópicos 8, 9 e 10, potenciará a aquisição do “saber fazer” e do pensamento crítico que são essenciais na aplicação da Estatística. O tópico 11 concentra-se num dos modelos fundamentais para a descrição experimental das leis da natureza que exprimem relações funcionais entre as quantidades medíveis relevantes e merece, por isso, tratamento diferenciado.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The objectives defined are classical ones for an introductory course in Probability and Statistics for Engineering degrees. These objectives have been fine-tuned from suggestions given by faculty from the scientific areas of the degrees. Topics 2, 3, 4 and 5 explain the essential concepts of Probability for future study of static models used in practice. In topic 1 language and basic notions of Probability are presented in order to create a context for further study. The sixth topic deals with one of the pillars of modern statistics not only justifying the prevalence of the normal model but also allowing the determination of confidence intervals and, by duality, the realization of hypothesis testing in the more relevant practical situations. Random number generation, in topic 7, is a basic tool in modern computation. The substantiation and study of a significant number of examples of parameter estimation and hypothesis testing, developed in the topics 8, 9 and 10, will potentiate the acquisition of know-how and critical thinking which are essential in the practical application of Statistics. Topic 11 concentrates on one of the fundamental models for the experimental description of the laws of nature expressing functional relations among measurable relevant quantities and deserves, by that reason, a separate treatment.

3.3.7. Metodologias de ensino (avaliação incluída):

Aulas teórico-práticas com exposição oral de matéria, com apresentação de exemplos e participadas com resolução de problemas. A avaliação consiste na realização de testes intercalares e de um exame final. Os testes darão origem a uma nota de avaliação contínua. Para obter a frequência o aluno tem que ter frequentado pelo menos três quartos das aulas.

3.3.7. Teaching methodologies (including evaluation):

Theoretical-practical classes with oral presentation of the matter, with examples and with problem solving, with student participation. The assessment consists on the realization interim tests and a final exam. The tests give rise to a continuous evaluation score. For the frequency the student must have attended at least two thirds of the classes.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Experiências pedagógicas relevantes levadas a cabo no âmbito da matemática a nível universitário e também nas Probabilidades e Estatística (ver i, ii e iii) referem a estratégia de associar a avaliação às actividades lectivas como eficaz. Tal justifica a componente da avaliação contínua por testes.

i- On the teaching and learning of Mathematics at University level, Pre-proceedings of the ICMI Study Conference, Singapore 1998.

ii- Dieudonné Leclerq (1998) Pour une pédagogie universitaire de qualité, , Mardaga.

iii- Steven G. Krantz (1999) How to Teach Mathematics, American Mathematical Society.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Relevant pedagogical experiences performed at university level in Mathematics and also in Probability and Statistics teaching (see i,ii, iii) refer that the strategy of associating evaluation with the lecturing-problem solving session is effective. This justifies the test continuous evaluation component.

i- On the teaching and learning of Mathematics at University level, Pre-proceedings of the ICMI Study Conference, Singapore 2008.

ii- Dieudonné Leclerq (1998) Pour une pédagogie universitaire de qualité, , Mardaga.

iii- Steven G. Krantz (1999) How to Teach Mathematics, American Mathematical Society.

3.3.9. Bibliografia principal:

- Guimarães, R. C. e Cabral, J. S. (1997).Estatística, McGraw-Hill
- Montgomery, D. C. e Runger, G. C. (2002).Applied Statistics and Probability for Engineers, 3ª ed., John Wiley and Sons
- Mood, A. M., Graybill e Boes (1974).Introduction to the Theory of Statistics. 3ª ed., McGraw-Hill
- Murteira, B., Ribeiro, C. S., Silva, J. A. e Pimenta, C., (2007).Introdução à Estatística, 2ª ed.McGraw-Hill
- Murteira, B. (1990).Probabilidades e Estatística, Vol. I e II, 2ª ed.,McGraw-Hill
- Paulino e Branco (2005).Exercícios de Probabilidade e Estatística. Escolar Editora
- Pedrosa, António (2004).Introdução Computacional à Probabilidade e Estatística.Porto Editora
- Pestana e Velosa (2006).Introdução à Probabilidade e à Estatística.Fundação Calouste Gulbenkian
- Rohatgi (1976).An Introduction to Probability Theory and Mathematical Statistics.Wiley
- Sokal e Rohlf (1995).Biometry.Freeman
- Tiago de Oliveira (1990).Probabilidades e Estatística:Conceitos,Métodos e Aplicações.,vol. I, II.McGraw-Hill

Mapa IV - Teoria da Computação / Theoretical Computer Science

3.3.1. Unidade curricular:

Teoria da Computação / Theoretical Computer Science

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (Responsável e Regente) (T-84h; OT-3h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Carla Maria Gonçalves Ferreira (P-84h; OT-3h)

Sofia Carmen Faria Maia Cavaco (P-84h; OT-3h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber: Usar a lógica matemática e teoria de conjuntos para modelar dados e sistemas simples. Conhecer conceito formal de algoritmo e a existência de problemas indecidíveis. Conhecer as classes de linguagens formais, os modelos computacionais associados e a sua relação mútua. Compreender a universalidade Turing e as suas consequências. Reconhecer a relevância prática da TC.

Saber Fazer: Modelar o espaço de estados de sistemas simples. Distinguir conjuntos contáveis de não contáveis. Definir a semântica de linguagens de expressões usando interpretações. Modelar sistemas simples usando autómatos (DFA e NFA). Construir DFA dada expressão regular e o inverso. Construir um DFA equivalente a um NFA dado. Construir um analisador lexical usando flex. Definir linguagens independentes de contexto usando gramáticas. Construir analisadores LL e LR. Construir analisador LL. Reconhecer a decidibilidade e indecidibilidade de problemas computacionais, ao nível adequado.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge: Use mathematical logic and set theory to model simple data and systems. Know the formal concept of algorithm, and the existence of undecidable problems. Know the fundamental classes of formal languages and associated computational models, and their relationship. Understand Turing universality and its consequences. Understand the practical relevance of TCS.

Application: Model the state space of simple systems. Distinguish between countable and non-countable sets. Define the semantics of simple expression languages by interpretation maps. Model simple systems using finite automata (DFA and NFA). Construct a finite automata from a regular expression and conversely. Construct a DFA equivalent to a given NFA. Construct a lexical analyser using flex. Define context free languages using grammars. Construct LL and LR parsers. Construct a LL using javacc. Recognize decidability and undecidability of computational problems, at the appropriate level.

3.3.5. Conteúdos programáticos:

1. Modelação com Conjuntos e Lógica

Conjuntos, Funções, Relações (revisão). Finito e Infinito, argumento diagonal de Cantor. Diferença entre função e algoritmo. Definições indutivas. Modelos de sistemas simples e tipos abstratos de dados.

2. Sintaxe e Semântica

Sintaxe concreta e sintaxe abstrata. Introdução informal às gramáticas (BNF). Modelos e Interpretações. Especificações algébricas.

3. Máquinas, Autómatos e Especificações

O que é um modelo computacional? Autómatos finitos deterministas e expressões regulares. Determinismo e não determinismo. Linguagens independentes de contexto e máquinas de pilha. Análise sintática (LL e LR).

4. Computabilidade

Complexidade básica (P, NP). Expressividade computacional. Equivalência entre programação funcional e imperativa. Máquinas abstratas e níveis de interpretação. Universalidade Turing. Tese de Church-Turing. Indecidibilidade (da terminação).

3.3.5. Syllabus:

1. Modeling with Sets and Logic

Sets, Functions, Relations (review). Finite and Infinite, Cantor's diagonalization argument. Difference between function and algorithm. Inductive definitions. Models of simple systems and abstract data types.

2. Syntax and Semantics

Concrete syntax and abstract syntax. Informal introduction to grammars (BNF). Models and interpretations. Algebraic specifications.

3. Machines, Automata and Specifications

What is a model of computation? Deterministic finite automata and regular expressions. Determinism and non-determinism in computation. Context Free languages and Stack machines. Parsing algorithms (LL and LR).

4. Computability

Basic complexity (P, NP). Expressiveness of computation models. Equivalence between pure functional and imperative programming. Abstract machines and layers of interpretation. Turing universality. Church-Turing thesis. Undecidability (of termination).

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Existe uma correspondência evidente entre os conteúdos e os objetivos.

Os objetivos de "Saber" são cobertos nos tópicos 1, 2, 3 e 4, com ênfase em 1, 2 e 4.

Os objetivos de "Saber fazer" são cobertos nos tópicos 1, 2, e 3 e 4 para o item "Reconhecer a decidibilidade e

indecidibilidade”.

3.3.6. Demonstration of the syllabus coherence with the curricular unit’s intended learning outcomes:

There is an evident correspondence between the syllabus and the curricular unit’s objectives

The “Knowledge” learning outcomes are covered in items 1, 2, 3 e 4, with an emphasis in 1, 2 and 4.

The “Knowledge Application” learning outcomes are covered in items 1, 2, e 3, with emphasis in 4 for item “Recognize decidability and undecidability”.

3.3.7. Metodologias de ensino (avaliação incluída):

O ensino da unidade curricular está organizado em aulas teóricas (2T) e práticas (3P). Existem notas, que seguem de perto os conteúdos das aulas teóricas.

Nas aulas práticas os alunos discutem e resolvem exercícios propostos pelo docente, de uma lista predefinida. Nas aulas teóricas são apresentados os conceitos e discutidas situações problemáticas, em geral motivadas por desafios gerais de várias áreas da informática. As competências de saber fazer são também exercitadas nas aulas teóricas, de forma a aumentar a ligação entre os conceitos teóricos e a sua aplicação.

A avaliação contínua consiste em dois testes (um intermédio, outro no fim do semestre) e 4 trabalhos práticos (TPs) , dos quais os alunos podem realizar apenas 3. Os TPs são realizados através de uma plataforma de e-learning (moodle). Os alunos que tenham reprovado na avaliação contínua com uma nota igual ou superior a 8 têm acesso ao exame final, cuja nota combinada com a dos trabalhos práticos produz a nota final.

3.3.7. Teaching methodologies (including evaluation):

The course is organized in recitation lectures (2T) and lab classes (3P). There are lecture notes, closely following the presentation in recitation.

In the lab classes students discuss and solve problems proposed by the instructor from a list. In the recitation lectures the instructor presents and motivates concepts and applications are discussed and exemplified, generally prompted by challenges arising in areas of computer science. “knowledge application” learning outcomes are also exercised in the recitation, to promote a close connection between the theoretical concepts and their application.

Evaluation consists in midterm and final tests and 4 individual assignments (students are required to submit at least . The 4 assignments are answered through an elearning platform (moodle). Any student that fails in the basic evaluations scheme with a grade equal of higher than 8/20 is admitted to final exam, which combines with the grading of the assignment to produce the final grade.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os trabalhos práticos permitem prender os alunos a um ritmo forte de acompanhamento da matéria, e a escolha de exemplos adequados nas aulas teóricas, retirados de situações práticas, facilita a motivação dos alunos para os fundamentos teóricos da computação, principalmente no capítulo da modelação com conjuntos. A metodologia seguida, apelando à intuição e esclarecendo bem a utilidade dos resultados teóricos, tem dado bons resultados, sendo a disciplina bastante apreciada pelos alunos, a julgar pelos resultados dos inquéritos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit’s intended learning outcomes:

The practical individual work assignments contribute to bind the students to a strong work rythm while following the course presentation. Chosen appropriate examples, inspired in practical problems, facilitates the motivation of students towards the theoretical foundations of computer science, in particulat in the modelling part. The methodology we have devised, which appeals to intuition and clarifies at each step the usefulness of the presented theoretical results seems to have provided good results. Previous versions of the course seems to have been fairly appreciated by the students, as witnessed by the results of surveys.

3.3.9. Bibliografia principal:

Lecture notes for the UC ITC (Luís Caires, 2011).

Christos Papadimitriou and Harry Lewis: “Elements of the theory of computation”, Prentice-Hall, 1982, second edition 1997.

Mapa IV - Bases de Dados / Databases

3.3.1. Unidade curricular:

Bases de Dados / Databases

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Júlio Alves Alferes (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Alexandre Carvalho Pinheiro Leite (Regente) (T-84h)

Joaquim Francisco Ferreira da Silva (P-112h)

Francisco de Moura e Castro Ascensão de Azevedo (P-112h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Arquitetura de um Sistema de Gestão de Bases de Dados*
- *Modelo de Entidades e Relações, o Modelo Relacional, e os fundamentos do Modelo Objeto/Relacional*
- *Álgebra Relacional*
- *SQL*
- *Teoria do desenho de bases de dados relacionais através da normalização com base em dependências funcionais e multivalor*
- *Modelação de dados em XML, e mecanismos para manipulação de dados semiestruturados*
- *Outras linguagens de consultas (Datalog e QBE)*

Saber Fazer:

- *Modelar, primeiro em ER e depois numa base de dados relacional, um problema de dimensão média*
- *Implementar, em SQL, uma base de dados relacional de tamanho médio, incluindo todos os mecanismos necessário à integridade dos dados*
- *Implementar uma interface simples para manipulação de uma base de dados*
- *Formular consultas complexas em Álgebra Relacional e SQL*
- *Usar os mecanismos base SQL do Modelo Objeto/Relacional*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Architecture of a Database Management System*
- *Entity-Relationship Model, Relational Model, and foundations of the Object-Relational Model*
- *Relational algebra*
- *SQL*
- *Database Design Theory through normalization using functional and multivalued dependencies*
- *Data modelling in XML and manipulation mechanisms for semisstructured data*
- *Other query languages (Datalog and QBE)*

Application:

- *Model a medium sized real problem, first using Entity-Relationship Diagrams and then a relational database*
- *Create a database in SQL for a medium-sized problem, including implementation of all mechanisms for guaranteeing data integrity*
- *Implement a simple interface for manipulation of the database*
- *Write complex queries in SQL and Relational Algebra*
- *Use the basic object-relational mechanisms of SQL*

3.3.5. Conteúdos programáticos:

- 1. Introdução aos sistemas de bases de dados*
- 2. Modelos de dados*
 - 2.1. Modelo ER*
 - 2.2. Modelo Relacional*
 - 2.3. Álgebra relacional*
- 3. Linguagens de manipulação de bases de dados*
 - 3.1. Linguagem SQL*
 - 3.2. Outras linguagens*
 - 3.3. QBE e Datalog*
- 4. Integridade e Segurança de Bases de Dados*
 - 4.1. Integridade de referência*
 - 4.2. Asserções e triggers*
 - 4.3. Transações*
 - 4.4. Segurança e autorizações*
- 5. Normalização de Bases de Dados*

- 5.1. *Dependências funcionais e multivalor*
- 5.2. *Formas normais: 3ª, 4ª e de Boyce-Cood*
- 6. *Discussão de outros modelos de bases de dados*
 - 6.1. *Bases de dados objetos/relacional*
 - 6.2. *Bases de dados dedutivas*
 - 6.3. *XML*

3.3.5. Syllabus:

- 1. *Introduction to Database Management Systems*
- 2. *Data models*
 - 2.1. *Entity-Relationship model*
 - 2.2. *Relational model*
 - 2.3. *Relational algebra*
- 3. *Database manipulation languages*
 - 3.1. *SQL query and manipulation language*
 - 3.2. *Other languages*
 - 3.3. *QBE and Datalog*
- 4. *Database integrity and security*
 - 4.1. *Referential integrity*
 - 4.2. *Assertions and triggers*
 - 4.3. *Transactions*
 - 4.4. *Security and authorizations*
- 5. *Relational databases normalization*
 - 5.1. *Functional and multivalued dependencies*
 - 5.2. *Normal forms: 3rd, 4th and Boyce-Codd*
- 6. *Discussion about other database models*
 - 6.1. *Object/relational databases (and the SQL case).*
 - 6.2. *Deductive databases*
 - 6.3. *XML*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos programáticos seguem de perto os objetivos de conhecimento estabelecidos, sendo assim clara a coerência entre ambos.

As aptidões objetivo da unidade curricular decorrem da execução do programa em atividades práticas laboratoriais. Assim, os conteúdos programáticos relativos aos modelos de dados são coerentes com o objetivo de modelação de um problema; os conteúdos relativos à linguagem SQL coerentes com o objetivo de implementação de uma base de dados relacional, os conteúdos relativos à Álgebra Relacional e SQL coerentes com o objetivo de formulação de consultas complexas; e os conteúdos sobre outros modelos de bases de dados coerentes com o objetivo da utilização dos mecanismos base do SQL do modelo Objeto/Relacional. A parte relativa a outras linguagens, incluindo de interface, é coerente com correspondente objetivo.

Todas as competências são também fomentadas pela forma de avaliação da unidade curricular, quer pelo exame quer pelo trabalho prático.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus follows very closely the knowledge learning outcomes established for the course, so it is clear the coherence between the two.

The know-how learning outcomes stem from the exercise of the program in lab sessions. Accordingly, the syllabus items corresponding to the Data Models are coherent with the modelling of a problem know-how objective; the items corresponding to the SQL coherent with the know-how of implementing a database; the items relative to SQL and Relational Algebra coherent with the know-how outcome of writing complex queries in both languages; the items relative to other database models coherent with the know-how outcome of using object-relational mechanisms of SQL. The items relative to other languages, including interface languages, are coherent with the corresponding outcome.

All learning outcomes are promoted through the evaluation mechanisms, both by the tests/exam and by the projects.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas e práticas. Nas primeiras são lecionados os conceitos e técnicas relevantes.

Nas aulas práticas são resolvidos problemas, feitas experiências em laboratório de computadores, e (parcialmente) desenvolvido o trabalho prático usando o Oracle.

A avaliação de conhecimentos inclui:

- *2 testes individuais teóricos (ou um exame de recurso), onde é avaliado o conhecimento que os alunos*

adquiriram dos conceitos e características das técnicas lecionadas.

• 1 trabalho prático, elaborado em grupo de 3 estudantes, que consiste no desenho, análise, e implementação de uma base de dados de pequena/média dimensão, incluindo a sua interface.

A entrega e avaliação do trabalho prático é feita em duas fases: numa primeira fase é entregue o desenho da base de dados; numa segunda fase é entregue o projeto completo. A avaliação do projeto final inclui uma apresentação/discussão.

3.3.7. Teaching methodologies (including evaluation):

The syllabus is taught in theoretical and laboratory classes. In the former, the main concepts and techniques are addressed.

The laboratory classes are dedicated to solving problems, and experimenting the various concepts of the syllabus using the DBMS Oracle, and (partially) developing the project.

The evaluation includes:

• 2 midterm individual tests (or an exam) where the knowledge acquired by the students on the concepts and techniques is assessed.

• 1 practical project, developed in groups of 3 students, consisting of the design, analysis and implementation of a small/medium size database, including its interface.

The project is developed/evaluated in two steps: first the design of the database; second, the complete project.

The evaluation of the final project includes a presentation/discussion.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular identificam a) a aquisição de conhecimento sobre desenho, análise, implementação e manipulação de bases de dados relacionais; b) a aplicação prática dos conhecimentos adquiridos sobre bases de dados relacionais; e c) aquisição de conhecimentos sobre outros modelos de dados. O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição do Modelo de Entidades e Relações, Modelo Relacional, Teoria da Normalização e SQL.

O objetivo b) é fundamentalmente atingido nas aulas práticas com a resolução de exercícios sobre modelação com o Modelo de Entidades e Relações, sua conversão para o Modelo Relacional, SQL, quer como linguagem de implementação de bases de dados, quer como linguagem de consultas, e normalização de Bases de Dados. O trabalho prático tem um papel fundamental na aplicação integrada de todas estas competências, no desenho, análise e implementação de uma base de dados de pequena/média dimensão.

O objetivo c) é adquirido nas aulas teóricas com a exposição do modelo Objeto-Relacional e do XML.

As matérias relativas aos objetivos a) e c) são avaliadas em testes individuais. As matérias relativas ao objetivo c) são avaliadas no trabalho prático.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The objectives of the course identify a) the acquisition of knowledge about design, analysis, implementation and manipulation of relational databases, b) the practical application of acquired knowledge on relational databases, and c) acquisition of knowledge about other data models.

Objective a) is mainly achieved in lectures where the following concepts are presented: Entity-Relationships Data Model, Relational Model, Normalization Theory and SQL.

Objective b) is primarily achieved in practical classes with the resolution of exercises on modeling with the Entity-Relationships Data Model, its conversion to the Relational Model, SQL, both as the implementation language of databases, as well as a query language, and normalisation of databases. The practical work has a key role in the integrated application of all these skills in the design, analysis and implementation of a database of small / médium size.

The objective c) is acquired in lectures with the presentation of the Object-Relational Data Model and XML.

Matters relating to the objectives a) and c) are evaluated through individual tests. Matters relating to objective c) are evaluated through the practical project.

3.3.9. Bibliografia principal:

Livros de Texto / Text Books

• Database System Concepts. A. Silberschatz, H. Korth and S. Sudarshan, McGraw-Hill 2010.

• Database Systems: the complete book. H. Garcia-Molina, J Ulmann and J. Widom. Prentice Hall, 2009

Manuais /Manuals

• Oracle 11g Documentation

3.3.1. Unidade curricular:

Linguagens e Ambientes de Programação / Programming Languages and Environments

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caíres (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Artur Miguel de Andrade Vieira Dias (Regente) (T-42h; P-84h; OT-3h)

Fernando Pedro Reino da Silva Birra (P-112h; OT-3h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Esta disciplina estuda as alternativas no desenho das linguagens de programação; compara e exercita algumas linguagens e paradigmas de programação relevantes; explica as técnicas gerais usadas no suporte à execução de programas.

Saber:

1 - Perceber as características e os cenários de utilização de um conjunto de linguagens de programação modernas: Caml, JavaScript, C, C++, Java.

2 - Compreender a funcionalidade e a arquitetura dos ambientes de suporte à programação e execução de programas.

3 - Possuir uma visão global dos aspetos envolvidos desde a especificação até à execução de programas

Saber Fazer:

4 - Resolver problemas pequenos, mas característicos, usando bem as linguagens de programação abordadas, dentro dum ambiente de desenvolvimento moderno

5 - Identificar as abstrações comuns e as diferenças de fundo entre diferentes linguagens de programação

6 - Expressar ou aproximar os mecanismos disponíveis numa dada linguagem de programação numa outra que não os possua como primitivos

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This course studies the various options in the design of programming languages; compares and applies some noteworthy languages and programming paradigms; explains the techniques used in programming execution environments.

Knowledge:

1 - Understand the characteristics and usage scenarios of a set of programming languages: Caml, JavaScript, C, C++, Java.

2 - Understand the functionality and the architecture of the environments for programming development and execution.

3 - General understanding of all the issues involved from the specification to the execution of programs.

Application:

4 - Solve small, but characteristic, problems in the programming languages addressed in the course.

5 - Identify common abstractions and the fundamental differences on how those abstractions are supported in different programming languages.

6 - Express or approximate mechanisms available in a given programming language using another language that does not have them as primitive.

3.3.5. Conteúdos programáticos:

A – Ambientes de programação e execução

Interpretação. Máquinas Virtuais. Níveis de interpretação.

Compilação. Ligação. Carregamento. Modelo de execução.

Portabilidade. Segurança.

Ferramentas: JDK; OCaml tools; Gnu tools; Cvs; Eclipse IDE.

B - Linguagens procedimentais e funcionais

OCaml e C.

Tipos: seu papel; verificação e inferência; polimorfismo; sobrecarga; tipificação estática e dinâmica.

Estruturação em blocos: ligações; ambientes; resolução de nomes.

Procedimentos e Funções: nomes locais e globais; parametrização; closures.

Gestão de Memória: manual e automática.

C - Linguagens baseadas em objetos

C++, JavaScript e Java.

Classes e protótipos.

Ligação dinâmica. Ocultação de informação. Subtipificação. Herança.

D - Linguagens de scripting e domínios específicos

JavaScript e Bash.

E - Técnicas de programação

Paradigmas de programação.

Método indutivo em OCaml. Manipulação de apontadores em C. Modelação em C++ e JavaScript.

F – Escolha duma linguagem

Fatores a considerar. Cenários.

3.3.5. Syllabus:

A – Programming and execution environments

Interpretation. Virtual Machines. Levels of interpretation.

Compilation. Linking. Loading. Execution model.

Portability. Security.

Tools: JDK; OCaml tools; Gnu tools; Cvs; Eclipse IDE.

B – Procedural and functional languages

OCaml and C.

Types: their role; type checking and inference; polymorphism; overloading; static and dynamic typing.

Block structuring: bindings; environments; name resolution.

Procedures and Functions: local and global names; parameterization; closures.

Memory management: manual and automatic.

C – Object-oriented languages

C++, JavaScript and Java.

Classes and prototypes.

Dynamic binding. Information hiding. Subtyping. Inheritance.

D – Scripting languages and domain-specific languages

JavaScript and Bash.

E – Programming techniques

Programming paradigms.

Inductive method in OCaml. Handling pointers in C. Modeling in C++ and JavaScript.

F – Selecting a programming language

Factors to consider. Scenarios.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O programa inclui os conceitos fundamentais que permitem compreender algumas das principais linguagens de programação modernas ao nível da sua semântica informal e da pragmática da sua implementação e utilização.

O objetivo 1 é coberto da seguinte forma: as características das linguagens são abordadas nos pontos B, C, D e E; os cenários de utilização são discutidos no ponto F.

Os objetivos 2 e 3, relacionados com o suporte à programação e execução, são endereçados pelo ponto A do programa.

O objetivo 4, de saber usar bem as linguagens estudadas, é coberto de forma direta pelo ponto E, mas também de forma transversal pelo resto do programa. A realização pelos alunos de 3 projetos de programação é um fator essencial para alcançar este objetivo.

Os objetivos 5 e 6, relacionados com a semântica informal das linguagens, são cobertos pelos pontos B, C e D.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus includes the fundamental concepts that support the understanding of major modern programming languages at the level of informal semantics and pragmatics of their use.

The objective 1 is covered as follows: the characteristics of the languages are discussed in the topics B, C, D and E; the usage scenarios are discussed in the topic F.

The objectives 2 and 3, about programming and execution support, are addressed by the topic A.

The objective 4, on using well the studied languages, is covered directly by topic E, but also across the board by the rest of the syllabus. The development of three programming projects by the students is an essential factor to

reach this goal.

Objectives 5 and 6, related to the informal semantics of the language, are covered by the topics B, C and D.

3.3.7. Metodologias de ensino (avaliação incluída):

Esta cadeira tem um forte carácter aplicado e cerca de 70% da nota dos testes e exames depende da capacidade do aluno usar bem as linguagens estudadas para resolver problemas de programação.

Nas aulas teóricas, os conceitos fundamentais da cadeira são transmitidos, exemplificados e discutidos. Nas aulas práticas, os alunos resolvem pequenos problemas onde aplicam os conceitos e técnicas estudados. Alguns desses problemas têm carácter teórico, mas a maioria são problemas de programação.

A avaliação consiste em dois testes (cada um com peso de 27.5%) e em três trabalhos de grupo de 2 alunos (trabalhos de dimensão pequena/média, cada um com peso de 15%). A nota de frequência é a média das notas dos três trabalhos. Para o aluno obter aprovação, precisa de nota de frequência positiva e que a média dos dois testes seja positiva. Se o aluno for a exame de recurso (para obter aprovação ou melhorar a nota), o exame substitui os dois testes nas contas anteriores.

3.3.7. Teaching methodologies (including evaluation):

This course has a strong applied approach and about 70% of the grade of the tests and exams depends on the student's ability to use well the studied languages in solving programming problems.

In the lectures, the concepts are presented, discussed and exemplified. In the lab classes, the students solve small problems, applying the concepts and techniques learned. Some of these problems are theoretical, but most are programming problems.

The assessment consists of two tests (each weighing 27.5%) and three projects for teams of two students (small/average sized projects, each weighing 15%). The attendance grade is the average of the grade of the three projects. To pass, the student must have a minimal attendance grade of 9.5, as well as a minimum of 9.5 on the average grade of the two tests. If the student take a ressit exam (either for passing or to improve the grade), this exam replaces the two tests in the previous rules.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas teóricas os alunos são iniciados nos conceitos e técnicas da cadeira. Pretende-se promover a construção do conhecimento e o desenvolvimento de alguma capacidade de análise crítica.

A solidificação dos conhecimentos, aptidões e competências ocorre de forma mais essencial nas aulas práticas e também fora das aulas durante a execução dos trabalhos da cadeira. Os exercícios e trabalhos cobrem praticamente toda a matéria e incluem desafios que conduzem os alunos a compreender melhor os conceitos e a usá-los de forma apropriada.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In the lectures, the students are initiated into the concepts and techniques of the course. There is also the goal promoting the desired knowledge construction and the development of some criticism capacity.

The consolidation of knowledge, abilities and skills occurs most essentially in the lab classes and also outside the classes during the development of the programming projects. The exercises and the projects cover almost the entire contents of the course; some of these exercises and projects include challenges that lead students to a better understanding of the concepts and to use them appropriately.

3.3.9. Bibliografia principal:

Artur Miguel Dias, Course notes, including exercises.

Jason Hickey, "Introduction to the Objective Caml Programming Language", 2006.

Brian Kernighan, Dennis M. Ritchie, "The C Programming Language", 2nd ed., 1988.

Bruce Eckel, "Thinking in C++, volume one", Prentice Hall, 2000.

Simon Willison, "A (Re)-Introduction to JavaScript", 2006.

John C. Mitchell. Concepts in Programming Languages. Cambridge University Press, 2003.

Ryan Stansifer. The Study of Programming Languages. Prentice-Hall International, Inc., 1995.

Mapa IV - Ciência, Tecnologia e Sociedade / Science, Technology and Society

3.3.1. Unidade curricular:

Ciência, Tecnologia e Sociedade / Science, Technology and Society

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Maria Paula Pires dos Santos Diogo (Responsável e Regente) (TP-32h; S-8h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Ana Maria de Oliveira Carneiro (TP-32h; S-8h)

José Luís Câmara Toivola Leme (TP-32h; S-8h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Objetivos da unidade curricular:

i) levar os alunos a interrogarem-se sobre a natureza e a extensão das relações entre ciência, tecnologia e sociedade no mundo actual;

ii) catalisar a reflexão crítica dos alunos sobre a sua futura experiência profissional e de cidadania.

iii) aumentar a capacidade de decisão e adaptação dos alunos num mundo em mudança.

Pretende-se:

i) aquisição de conhecimentos:

- compreender a estrutura da tecnociência e sua relação com os contextos económico, político, social e cultural;
- dominar conceitos fundamentais para a análise das inter-relações entre ciência, tecnologia e sociedade.

(ii) aquisição de competências:

- perspetivar o relacionamento entre ciência e a tecnologia e sociedade;
- construir uma memória crítica sobre o papel da ciência e da tecnologia na sociedade europeia;
- desenvolver o sentido de ética e responsabilidade social do cientista e do engenheiro;
- relacionar a prática profissional com a prática de uma cidadania crítica e consciente.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This course aims at:

(i) leading students to ask themselves crucial questions on the nature of the relationship between science, technology and society;

(ii) leading students to think about their future work as engineers and about their rights and duties as citizens;

(iii) increasing the students' capacity of decision and adjustment in a changing world.

(i) specific capabilities to be developed:

- to understand the structure of technoscientific knowledge and its relations with social, economic, and cultural contexts;

- to master the fundamental concepts for the analysis of the interrelationship between science, technology and society.

(ii) general capabilities to be implemented:

- to understand the dynamics of the relationship between science, technology and society;

- to build a critical memory on the role of science and technology in European society;

- to develop a sense of ethics and social responsibility;

- to relate professional practice with the with active citizenship.

3.3.5. Conteúdos programáticos:

1 Ciência, Tecnologia e Sociedade

1.1 Tópicos fundamentais na análise da relação ciência, tecnologia e sociedade

1.2 Ética, responsabilidade social e cidadania

1.3 A Sociedade da Informação e a contemporaneidade.

2 Risco, Segurança e Responsabilidade: sociedade de risco e a ética moderna.

3 Ciência, Tecnologia e Género: as mulheres no trabalho em ciência e tecnologia; o género na construção do discurso científico.

4 Redes de Sustentabilidade - ambiente e sociedade: intersecções entre decisão política/económica, competências científicas e técnicas e questões ambientais.

5 Guerra e Paz: Einstein, Bohr e Oppenheimer: modelos de investigação tecnocientífica contemporâneos e responsabilidade social.

6 O Futuro Bio e Nano: landmarks e debates políticos e éticos.

7 E o Homem Criou o Ciborgue: ciência, tecnologia e cultura popular; medos e desconfianças; fronteiras entre humano e não-humano.

3.3.5. Syllabus:

1.Science, Technology and Society

1.1 Critical issues of the relationship between science, technology and society

1.2 Ethics, social responsibility and citizenship

1.3 The Information Society.

2.Risk, Safety, Responsibility and Accountability: risk society and modern ethics.

3.Science, Technology and Gender: women in science and technology; gender issues in the construction of scientific discourse.

4.Sustainability Networks - Environment and Society: intersections between political/economic decisions, scientific and technical expertise and environmental issues.

5.War and Peace: Einstein, Bohr and Oppenheimer: models of contemporary techno-scientific research and social responsibility.

6.The Bio and Nano Future: landmarks and ethical debates.

7.And Man Created Cyborg: science, technology and pop culture; fears and distrust; the thin line between human and nonhuman.

8.Making Modernity Visible. Science, Technology and Cinema: film narrative and technoscience.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular são suscitar uma reflexão sobre a sociedade tecnocientífica contemporânea, proporcionar uma abordagem integrada dos problemas actuais simultaneamente na perspectiva do produtor de saberes e do cidadão utilizador e assegurar os “utensílios” para melhorar as capacidades de decisão,numa sociedade em mudança acelerada.

O programa organiza-se em módulos que abordam tópicos fundamentais e exemplares para uma problematização crítica do relacionamento entre ciência,tecnologia e sociedade, nos dias de hoje,caracterizados pela mudança rápida e pela globalização.

Parte-se da experiência dos alunos,moldada por um conjunto de conteúdos específicos em interrogação/reflexão crítica. A partir desse conjunto, usamos exemplos para ilustrar as dinâmicas de desenvolvimento e as diferentes soluções encontradas para as interrogações iniciais.Em todos os módulos as questões da ética, da cidadania e da percepção pública da ciência e da técnica têm um papel fundamental.

3.3.6. Demonstration of the syllabus coherence with the curricular unit’s intended learning outcomes:

The aims of this course are to lead students to understand nowadays technoscientific society, to provide an unified approach about science, technology and society, using both the perspective of the professional and of the citizen and to ensure that students develop the basic skills to make decisions in a fast changing society. The program is organized into modules that address key topics concerning the relationship between science, technology and society and present examples that are used to stimulate the students’ critical reasoning about nowadays society, characterized by rapid change and globalization.

The program builds on students’ individual experiences reshaping them as general questions about the nature of the relationship science/technology/society. Key examples are used to illustrate this complex relationship, leading students to understand it as a dynamic process. In all modules ethics, citizenship, and public perception of science and technology play a crucial role.

3.3.7. Metodologias de ensino (avaliação incluída):

Cada sessão da unidade curricular (UC) tem duas horas teórico-práticas, onde a exposição dos conteúdos do programa são assegurados pelo docente. Cada módulo é constituído por 2 aulas (4 h). Destas,1h:30m são dedicados ao trabalho sobre materiais didácticos complementares relevantes, nomeadamente iconografia diversa,extratos de obras científicas,técnicas e de literatura,em ambos os casos coevas da matéria lecionada na sessão,e filmes.

Procura-se sempre estimular nos alunos uma leitura crítica e integrada destes materiais didáticos nos conteúdos do programa através dos quais serão directamente avaliados

A UC é avaliada por 4 mini-testes e pela realização de 2 quizzes.Os mini-testes têm como objectivo a avaliação da compreensão e domínio dos conteúdos do programa da disciplina;os quizzes respeitam ao trabalho de interpretação do material de apoio apresentado nas sessões em novos contextos, de forma autónoma.Uma terceira componente é a intervenção nos debates suscitados nas aulas.

3.3.7. Teaching methodologies (including evaluation):

Each session lasts two-hours (theory and practice) and each module is composed by two sessions (4 hours). The contents of the program are presented by the teacher and supported by slides, technical texts, literature, and films covering the topics outlined in the syllabus. About one hour and a half should be dedicated to reading

and audio-visual didactic contents.

Students are encouraged to have a critical posture concerning the topics of the program.

The course is evaluated by 4 mini-tests and 2 quizzes. The mini-tests aim at assessing if students master the contents of the syllabus; the quizzes aim at assessing if students are able to use what they learned in classes in different learning environments and in an autonomous way. Students are also evaluated by their participation in the debates that take place during the sessions.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Sendo o objetivo da unidade curricular suscitar uma reflexão sobre a sociedade contemporânea, moldada pela tecnociência, proporcionar uma abordagem integrada que permita equacionar os problemas atuais simultaneamente na perspetiva do produtor de saberes e do cidadão utilizador e assegurar os “utensílios” básicos para melhorar as capacidades de decisão, numa sociedade em mudança acelerada, as metodologias de ensino têm de ter, obrigatoriamente, uma componente teórica e uma prática.

Com a componente teórica pretende-se que os alunos adquiram uma série de conhecimentos de forma fundamentada e afastando-se do senso-comum nestas matérias. Cabe ao professor apresentar estas matérias aos alunos, privilegiando sempre uma abordagem reflexiva que tenha em conta que o que se pretende é incutir nos alunos uma postura crítica e interrogativa e não um acumular gratuito de conhecimentos.

A utilização de materiais de apoio como imagens, textos literários e filmes permite, por um lado, tornar mais acessíveis os conteúdos através da visualização, e, por outro, mostrar aos alunos que a ciência e a tecnologia não vivem ao lado da sociedade, mas são parte dela, moldando-a e sendo moldadas.

Paralelamente às sessões os alunos têm um conjunto de complementar de materiais de trabalho que devem ver e analisar sem a presença do professor. O objectivo desta componente é desenvolver a autonomia dos alunos, permitindo-lhes trabalhar fora das aulas, por si próprios, embora com um objectivo pré-definido.

A preparação para estes trabalhos pode ser feita individualmente ou em grupo, apesar do trabalho em si, pela forma como os quizzes estão concebidos, ser necessariamente individual.

Assim, em ambiente de aula, os alunos adquirem conhecimentos e debatem em grupo elementos didácticos complementares; com o trabalho se a presença do professor, encorajam-se nos alunos os factores de autonomia e de responsabilidade e, em muitos casos, de entreajuda.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

This course aims at leading students to understand nowadays society, deeply shaped by technoscience, at providing an approach that thinks about science, technology and society in an unified way, using both the perspective of the professional and of the citizen (producers'/consumers' perspectives) and at ensuring that our students develop the basic skills to make decisions in a fast changing society. Bearing in mind these objectives, the teaching methodologies require both a theoretical and a practical approach, thus offering the opportunity to ground theory in real examples.

During classes the students acquire a solid knowledge on the topic of the role of technology in the making of Europe, avoiding the common sense perspective in these matters. The teacher presents the contents of the program, always giving priority to a critical approach that encourages students to think rather than just to listen in a passive way.

By using images, literary texts and films during the sessions we aim at, on one hand, making contents more accessible to students, and, on the other hand, to show that technology does not live next to society, but it is part of society, thus changing and being changed in this mutual relationship.

Alongside the sessions the students have to do some “homework”, by analysing a set of complementary materials. The objective of this work is to develop students' autonomy, allowing them to work outside classes by themselves, albeit with a pre-defined objective.

The preparation for this “homework” can be done individually or in groups, although the work itself, due to the quizzes' design, is mandatorily individual.

Thus, in the classroom environment, students acquire knowledge and discuss critical issues among them with the teacher's guidance; outside classes, by themselves, individually or in group, students discuss and analyse new case-studies, thus exercising their knowledge and developing personal skills such as autonomy, responsibility and mutual help.

3.3.9. Bibliografia principal:

*Allhoff, F. et al (eds.), Nanoethics: The Ethical and Social Implications of Nanotechnology, Wiley, Hoboken, ,2007.
Brodwin, P.E. (ed.), Biotechnology and Culture: Bodies, Anxieties, Ethics, Indiana University Press, Bloomington, 2000.*

Carson, R., Silent Spring, Boston, Houghton Mifflin Company, 1962.

Castells, M., Rise of The Network Society, Londres, Blackwell Editors, 1996.

Collins, H., Pinch, T., The Golem at Large, Cambridge, Cambridge University Press, 1998.

Irwin, A., Sociology and the Environment, Polity Press, Cambridge, 2001.

Jonas, H., The Imperative of Responsibility: In Search of Ethics for the Technological Age, University of Chicago Press, Chicago, 1984.
Evetts, J., Gender and Career in Science and Engineering, Londres, Taylor and Francis, 1996.
Malartre, E., Benford, G., Beyond Human: Living with Robots and Cyborgs, Nova Iorque, Forge Books/Macmillan, 2007.
Goodchild, P., J. Robert Oppenheimer, Shatterer of Worlds, Nova Iorque, Fromm Int. Publishing Corporation, 1980.

Mapa IV - Computação Gráfica e Interfaces / Computer Graphics and Interfaces

3.3.1. Unidade curricular:

Computação Gráfica e Interfaces / Computer Graphics and Interfaces

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Manuel João Toscano Próspero dos Santos (Responsável e Regente) (T-84h; OT-2h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Adriano Martins Lopes (P-56h; OT-2h)
João Miguel da Costa Magalhães (P-112h; OT-2h)
Teresa Isabel Lopes Romão (P-28h; OT-2h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber

- *Descrever o pipeline gráfico e identificar as respetivas implicações.*
- *Enumerar e descrever as técnicas básicas para modelação baseada em polígonos e atributos de vértices.*
- *Identificar os princípios de base na construção da interface com o utilizador.*

Saber Fazer

- *O aluno deverá ser capaz de usar uma API moderna para criar uma aplicação gráfica que possa ser integrada com aplicações de qualquer outro tipo.*
- *O aluno conseguirá traduzir os princípios de conceção duma interface gráfica com o utilizador numa implementação eficiente e com aplicabilidade real.*

Competências Complementares

- *Capacidade de modelação e abstração.*
- *Avaliação teórica da eficiência de uma solução.*
- *Elaboração e seguimento de um protocolo experimental para avaliação de uma solução.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge

- *To describe the graphics pipeline and to identify their implications.*
- *To list and describe the basic techniques for modelling based on polygons and attributes of vertices.*
- *To identify the basic principles in building the user interface.*

Application

- *The student should be able to use a modern API to create a graphical application that can be integrated with applications from any other kind.*
- *The student should manage to translate the principles of design of a GUI into an efficient implementation and actual applicability.*

Soft-Skills

- *Ability for modelling and abstraction.*
- *Theoretical assessment of the efficiency of a solution.*
- *Drafting and following an experimental protocol for evaluation of a solution.*

3.3.5. Conteúdos programáticos:

1. Modelação

Primitivas

Superfícies

Grafos de cena

2. Projeções geométricas

Projeções Paralela e Perspetiva

- 3. Outras transformações essenciais no pipeline gráfico*
- Enquadramento janela-visor*
- Recorte*
- Remoção de partes ocultas*
- 4. Programação por eventos*
- 5. Animação gráfica em tempo real*
- 6. Introdução aos modelos de cor e de iluminação*
- 7. Geração e aplicação de texturas por mapeamento*
- Imagens digitais e métodos procedimentais*
- 8. Interação*
- Modelos*
- Estilos*
- Design do ecrã*
- Contextos*
- 9. Técnicas de interface com o utilizador*
- Metáforas 2D/3D*
- Dispositivos físicos*
- Controlo*
- Feedback*
- Visualização*
- Navegação e Manipulação*
- 10. Programação em OpenGL*
- 11. Programação em X3D*

3.3.5. Syllabus:

- 1. Modeling*
- Primitives*
- Surfaces*
- Scene graphs*
- 2. Geometric projections*
- Parallel and perspective projections*
- 3. Other essential transformations in the graphics pipeline*
- Window-viewport transformation*
- Clipping*
- HLHSR*
- 4. Event-driven programming*
- 5. Computer animation in real time*
- 6. Introduction to color models and illumination models*
- 7. Generation and application of texture mapping*
- Digital images and procedural methods*
- 8. Interaction*
- Models*
- Styles*
- Screen design*
- Contexts*
- 9. User interface techniques*
- 2D/3D metaphors*
- Physical devices*
- Control*
- Feedback*
- Visualization*
- Navigation and manipulation*
- 10. OpenGL programming*
- 11. X3D programming*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O programa inclui todos os conceitos necessários à compreensão da funcionalidade de um pipeline gráfico (2, 3, 6 e 7). São lecionadas as transformações básicas das primitivas 2D/3D em pixels no ecrã (1 e 2). Também são estudadas diversas ferramentas matemáticas para a conceção de modelos, hierárquicos ou não (1), assim como algumas técnicas de interação relevantes (4, 5, 8 e 9). A inclusão de OpenGL e de X3D permite a aprendizagem de ferramentas atuais com grande utilidade prática na programação numa grande diversidade de aplicações (10 e 11).

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus includes all the concepts needed to understand the functionality of a graphics pipeline (2, 3, 6 and 7). The basic transformations of 2D/3D primitives into pixels on the screen are present (1 and 2). Several mathematical tools are also studied for the design of models, hierarchical or not (1), as well as some relevant interaction techniques (4, 5, 8 and 9). The inclusion of OpenGL and X3D allows learning current tools with great practical usefulness in programming of a wide variety of applications (10 and 11).

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas, as matérias são explicadas e ilustradas com exemplos de programação. Em geral, todos os resultados principais são mostrados com base numa variedade de meios, incluindo applets.

Nas aulas práticas, os alunos fazem pelo menos dois projetos em grupo que abrangem a programação de questões importantes em computação gráfica e onde eles projetam e implementam a interface gráfica para interagir com as aplicações gráficas 2D/3D correspondentes. Os alunos devem frequentar a maioria das aulas práticas.

A cada aluno será atribuída uma nota individual resultante da avaliação dos projetos de programação e de dois testes intermédios. Este procedimento irá substituir o exame final em caso de aprovação, mas o aluno deve obter uma nota mínima com os dois testes.

3.3.7. Teaching methodologies (including evaluation):

In theoretical classes, matters are explained and illustrated with programming examples. In general, all the main results are shown using a variety of means, including applets.

In practical classes, students do at least two group projects covering the programming of important issues in computer graphics and the design of a suitable GUI to interact with the corresponding 2D/3D graphical applications. Students must attend the majority of practical classes.

Each student will be assigned an individual grade resulting from the evaluation of the programming projects and two mid-term tests. This procedure will substitute the final exam in case of approval, but the student must achieve a minimum grade with the two mid-term tests.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Através das aulas teóricas o aluno é iniciado nos conceitos, técnicas, ferramentas matemáticas e algoritmos que permitem a compreensão da modelação de dados e visualização em qualquer aplicação gráfica interativa comum. A avaliação relacionada é baseada em dois testes intermédios.

As aulas práticas são utilizadas principalmente para a conceção e execução dos projetos, podendo os alunos fazer perguntas de esclarecimento. Esses projetos são aplicações gráficas 2D/3D interativas que usam uma API moderna.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Throughout lectures the student is introduced to the concepts, techniques, mathematical tools and algorithms allowing the understanding of data modeling and viewing within any common interactive graphical application. The related evaluation is based on two mid-term tests.

Practical classes are mainly used for the design and implementation of the projects on which students can ask questions for clarification. These projects are interactive 2D/3D graphical applications using a modern API.

3.3.9. Bibliografia principal:

- John F. Hughes, Andries van Dam, Morgan McGuire, David Sklar, James D. Foley, Steven K. Feiner, Kurt Akeley, "Computer Graphics: Principles and Practice", 3rd Edition, Addison-Wesley Professional (2013), ISBN-13: 978-0321399526
- J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, "Computer Graphics - Principles and Practice", 2nd edition in C, Addison-Wesley (1996), ISBN: 0201848406
- Donald Hearn, M. Pauline Baker, Warren Carithers, "Computer Graphics with OpenGL, 4th edition, Prentice Hall (2010), ISBN-13: 978-0136053583
- Don Brutzman, Leonard Daly, "X3D: 3D Graphics for Web Authors", Elsevier (2007), ISBN: 012088500X
- Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis (OpenGL Architecture Review Board), "OpenGL Programming Guide: The Official Guide to Learning OpenGL", 4th edition, Version 1.4, Addison-Wesley (2003), ISBN: 0321173481
- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "Human-Computer Interaction", 3rd edition, Prentice Hall (2004), ISBN: 0130461091

Mapa IV - Inteligência Artificial / Artificial Intelligence

3.3.1. Unidade curricular:

Inteligência Artificial / Artificial Intelligence

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Carlos Augusto Isaac Piló Viegas Damásio (Responsável e Regente) (T-42h; P-56h; OT-1h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Susana Maria dos Santos Nascimento Martins de Almeida (P-56h; OT-1h)

Jorge Carlos Ferreira Rodrigues da Cruz (P-56h; OT-1h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber

- *Paradigmas, áreas e aplicações da Inteligência Artificial.*
- *Algoritmos de procura cega, informada e local. Heurística e função de avaliação. Complexidade.*
- *Inferência em lógica proposicional, de 1ª ordem e em redes Bayesianas. Limitações.*
- *Linguagens de planeamento e algoritmos para construção de planos.*
- *Abordagens da aprendizagem automática e algoritmos essenciais.*

Saber fazer

- *Modelar problemas de procura, especificar heurísticas e avaliar o seu comportamento.*
- *Implementar algoritmos de procura. Parametrização e experimentação dos algoritmos.*
- *Modelar e resolver problemas com Programação em Lógica.*
- *Modelar conhecimentos com redes Bayesianas. Efetuar inferência preditiva e de diagnóstico.*

Competências complementares

- *Escolher apropriadamente instrumentos pesando a qualidade da solução e o tempo/espaco necessário para a obter.*
- *Justificar ou selecionar soluções.*
- *Trabalhar em equipa com a escrita de relatórios.*
- *Capacidade de modelação abstrata.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge

- *Paradigms, areas and applications of Artificial Intelligence.*
- *Blind, informed and local search algorithms. Heuristics and evaluation functions. Complexity.*
- *Inference in propositional logic, first order logic and Bayesian networks. Limitations.*
- *Planning languages and algorithms for plan generation.*
- *Learning approaches and its essential algorithms.*

Application

- *Model search problems, specify heuristics and assess its behaviour in practice.*
- *Implement search algorithms. Parameterise and experiment with the algorithms.*
- *Model and solve problems with Logic Programming.*
- *Model knowledge with Bayesian networks. Perform predictive and diagnosis tasks.*

Soft skills

- *Choose appropriately instruments weighing the solution quality against the time/space necessary to obtain it.*
- *Justify or choose solutions.*
- *Do team work with report writing and deadline fulfilment.*
- *Abstract modelling capabilities.*

3.3.5. Conteúdos programáticos:

Teóricas (42 horas):

1.Panorâmica e breve história da IA e suas aplicações (3h)

2.Resolução de problemas (12h)

1.Agentes e problemas de procura

2.Procura cega

3.Procura heurística em espaços de estados. Algoritmo A.*

4.Procura local e problemas de otimização

5.Agentes adaptativos

6.Problemas de Satisfação de Restrições

7.Procura com Adversários

3.Representação do Conhecimento e Raciocínio (12h)

1.Lógica Proposicional

2.Lógica de Predicados de Primeira Ordem

3.Programação em Lógica

4.Planeamento (4,5h)

1.Representação de problemas de planeamento clássicos

2.Linguagens e algoritmos de planeamento

3.GRAPHPLAN

5.Incerteza e raciocínio probabilístico (4,5h)

1.Teoria das Probabilidades e Regra de Bayes

2.Redes Bayesianas

6.Aprendizagem Automática (4,5h)

1.Abordagens ao problema da aprendizagem

2.Aprendizagem conceptual e indutiva

3.Redes Neurais

7.Conclusão (1,5h)

Práticas (26h): Trabalho Procura (10h), Trabalho Prog. em Lógica (10h), Redes de Bayes (2h), Exercícios (4h)

3.3.5. Syllabus:

Theory (45 hours - indicative)

1.Overview of Artificial Intelligence and its applications (3h)

2.Problem-solving (12h)

1.Search agents and search problems.

2.Blind search.

3.Heuristic search in state-space graphs. A* algorithm.

4.Local search and optimisation problems.

5.Adaptative agents.

6.Constraint Satisfaction Problems.

7.Adversarial search.

3.Knowledge Representation and Reasoning (12h)

1.Propositional logic.

2.First-order logic.

3.Logic programming

4.Planning (4.5h)

1.Logical representation of classical planning problems.

2.Planning languages and algorithms. Partial order planning.

3.GRAPHPLAN

5.Uncertainty and probabilistic reasoning (4.5h)

1.Probability theory and Bayes' rule.

2.Bayesian networks.

6.Machine Learning (4.5h)

1.Learning agents. Approaches to the learning problem.

2.Conceptual and inductive learning.

3.Neuronal networks.

7.Conclusion (1.5h)

Labs (26h): Search project (10h), Logic programming project (10h), Bayesian Networks (2h), Exercises (4h).

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular estão desenhados para dotar os alunos os conhecimentos e aptidões em tópicos centrais abordados pela Inteligência Artificial, nomeadamente Procura, Representação do Conhecimento e Raciocínio, Planeamento e Aprendizagem Automática. Esta seleção de tópicos cobre todos os aspetos referenciados como core do ACM-IEEE CS Curriculum 2008 e Core-Tier2 na proposta CS Curriculum 2013, complementados com tópicos “elective” para melhor compreensão e aprofundamento das matérias.

Os “saberes” são fornecidos essencialmente pela componente teórica do programa, sendo clara a sua coerência, enquanto os “saber fazeres” são obtidos com a realização dos 2 trabalhos práticos de grupo e participação nas aulas práticas. O primeiro trabalho prático é realizado em Java e o segundo em Prolog. O As competências obtêm-se da conjugação das componentes práticas e teóricas da unidade curricular, nomeadamente pela realização de relatórios de trabalho obrigatórios.

3.3.6. Demonstration of the syllabus coherence with the curricular unit’s intended learning outcomes:

The objectives of the unit are designed to provide the students with the knowledge and application outcomes in central areas of Artificial Intelligence, namely Search, Knowledge Representation and Reasoning, Planning and Machine Learning. This selection of topics covers the o ACM-IEEE CS Curriculum 2008 core body of knowledge and of CS Curriculum 2013 proposal Core-Tier2 hours, complemented with some electives for better understanding and coverage of topics.

The knowledge objectives are achieved mostly with the theoretical component of the program, being clear their coherence, while the application objectives are attained with the development of the 2 practical projects in teams and participation in lab classes. The first project is performed in Java and the second in Prolog. The competences are obtained from the interplay of the practical and theoretical components of the curricular unit, namely by delivering the mandatory project reports.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas são expostos oralmente os pontos essenciais da matéria que deverão ser complementados com a leitura do manual aconselhado. As aulas laboratoriais destinam-se à apresentação e realização dos trabalhos práticos sobre procura e representação do conhecimento, acompanhadas pelos docentes. Algumas das aulas práticas destinam-se à modelação e realização de exercícios selecionados de exames anteriores, em particular de redes Bayesianas.

Os alunos têm ao seu dispor uma página da cadeira que mantém informação atualizada sobre o funcionamento da cadeira, sendo disponibilizados acetatos da matéria teórica, enlaces para material na Web, guiões dos trabalhos práticos e soluções de exames.

A avaliação é contínua e composta por 2 trabalhos práticos de grupo (15% da nota cada), 2 testes (35% da nota final cada) durante o semestre ou um exame escrito individual final (70% da nota em substituição dos testes). É necessário ter média dos testes ou nota de exame superior a 9,5 valores.

3.3.7. Teaching methodologies (including evaluation):

The most important subjects will be presented orally in the lectures, complemented by reading of the reference book. The lab classes are devoted to the presentation and implementation of the programming projects about selected topics of the course: search project and knowledge representation project. Lecturers accompany the development of the projects. Some of the lab classes are devoted to the modelling and solving of selected exercises of previous midterms, particularly about Bayesian networks.

The students have access to slides used in the classes as well as programming project guidelines. There is also a course web page keeping updated information about the course, namely deadlines for programming projects, and links to material available in the Web.

Course evaluation is composed of 2 projects (15% of the grade each), 2 midterms (35% each) or a final written exam (70% substituting midterms). It is mandatory to have average in midterms or final exam ≥ 9.5 .

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Todos os tópicos mencionados nos objetivos “Saber” são tratados com grande detalhe nas aulas teóricas da unidade curricular, sendo aprofundados ainda mais nos capítulos do manual indicados para leitura. Os tópicos são apresentados e ilustrados com problemas concretos, com uma análise profunda das características, propriedades, vantagens e desvantagens das aproximações e respetivos algoritmos. São também mostrados vídeos e ferramentas explorando as aplicações reais da Inteligência Artificial e funcionamento dos algoritmos estudados. Estes tópicos são estudados a um nível de detalhe que permitem aos estudantes modelar e resolver pequenos problemas, apreendendo a dificuldade de resolução de problemas similares em aplicações realistas.

A componente “Saber fazer” é obtida pela realização de 2 trabalhos práticos. No primeiro deles são implementados diversos algoritmos de procura cega, informada e local para a resolução de um problema de navegação de um agente. Os alunos têm de definir heurísticas para o problema e compará-las experimentalmente na prática, assim como avaliar o desempenho dos algoritmos quer em termos temporais quer em termos de espaço. No segundo trabalho introduz-se uma linguagem de programação em lógica (Prolog) para representar conhecimento necessário à resolução de uma tarefa de planeamento, implementando também um planeador simples.

Os momentos de avaliação também se encontram organizados para reforçar a aprendizagem. Assim, tem-se um primeiro teste onde são avaliados os tópicos correspondendo essencialmente aos itens do programa 1, 2 e parte do 3 (lógica proposicional), seguido da entrega de trabalho. Os restantes tópicos são avaliados no final do semestre em alinhamento com a conclusão do segundo trabalho.

Existem ainda aulas práticas preparatórias para os trabalhos e testes, nas primeiras sendo ilustrados problemas e sua modelação. Nas aulas preparatórias para os testes são resolvidos exercícios de modelação que servem eles próprios para complemento quer do trabalho quer da matéria teórica.

As competências são obtidas essencialmente pela realização dos trabalhos em equipa, sua avaliação experimental, exposição dos resultados obtidos e sua justificação num relatório escrito, assim como pelos aspetos de modelação abstrata exigidos nos testes de avaliação. Os testes e exame encontram-se desenhados quer para avaliar os conhecimentos teóricos do aluno assim como as suas aptidões para modelar novos problemas, discutir e avaliar a sua solução.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

All the “Knowledge” topics are treated with great detail in the lectures of this curricular unit, and expanded in the selected chapters of the reference book to be read by the students. The topics are presented and illustrated with concrete problems, with an analysis of characteristics, properties, advantages and disadvantages of the approaches and corresponding algorithms. Videos and tools are also shown to provide real applications of Artificial Intelligence and behaviour of the studied algorithms. These topics are studied at a level of detail enabling the students to model and solve toy problems, realizing the difficulty of solving similar problems in realistic applications.

The “application objectives” are acquired by the development of two projects. In the first project, students implement search algorithms (blind, informed and local) and model a navigation problem of an agent. Students must define heuristics for the algorithms and compare them in practice, and additionally assess the behaviour of the algorithms in terms of time and space. In the second project, the students are introduced to a logic programming language (Prolog) in order to represent knowledge necessary to solve a planning problem, and implement a simple planning algorithm.

The evaluation moments are planned to reinforce the learning outcomes. The first midterm addresses the topics corresponding to modules 1, 2 and part of 3 (propositional logic) of the program, followed by the delivery of the project. The remaining topics are assessed at the end of the semester in alignment with the conclusion of the second project.

There are additionally preparatory lab classes for the projects and midterms, in the former being illustrated problems and their modelling. In the latter, some selected exercises are solved complementing the project and the theoretical subjects.

Competences are obtained essentially by the development of team projects, their experimental evaluations, exposure to the obtained results and justification in a written report, as well as the abstract modelling competences assessed in the midterms.

The midterms and exams are structured to evaluate both the knowledge of students, as well as the skills to model new problems, discuss and assess their solution.

3.3.9. Bibliografia principal:

O manual adotado nesta disciplina é [Stuart Russell & Peter Norvig, 2010].

Alguns tópicos específicos são somente tratados no manual em Português [Costa & Simões, 2008].

[Sterling & Shapiro, 1994] é a referência à linguagem Prolog. Os restantes livros são de leitura opcional.

Adotados/Adopted

•Stuart Russel and Peter Norvig. Artificial Intelligence. A Modern Approach, 3rd edition. Prentice-Hall, Inc., 2010.

•Ernesto Costa e Anabela Simões. Inteligência Artificial. Fundamentos e Aplicações. 2ª edição, FCA, 2008.

•Leon Sterling and Ehud Shapiro. The Art of Prolog (2nd Edition). MIT Press, 1994

Complementares/Complementary

•Matt Ginsberg. Essentials of Artificial Intelligence. Morgan Kaufmann Publishers, 1994

•Nils J. Nilson. Artificial Intelligence: A new synthesis. Morgan Kaufmann Publishers, 1998

•I. Bratko. Prolog Programming for Artificial Intelligence, 3rd Edition, Pearson-Education, 2001.

•P. Blackburn, J. Bos, e K. Striegnitz. Learn Prolog Now! College Publications, 2006.

Mapa IV - Métodos de Desenvolvimento de Software / Software Development Methods

3.3.1. Unidade curricular:

Métodos de Desenvolvimento de Software / Software Development Methods

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:
Ana Maria Diniz Moreira (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:
Vasco Miguel Moreira do Amaral (Regente)(T-42h; P-28h)
João Batista da Silva Araújo Júnior (P-168h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber

- a) Ciclo de desenvolvimento, ferramentas, métodos e técnicas de gestão, especificação e desenho de projetos em larga escala, que envolvam equipas grandes e multidisciplinares*
- b) Importância das atividades de desenvolvimento de um projeto, das abordagens e dos modelos que integram estas atividades para enfrentar problemas complexos de uma forma pragmática, mas rigorosa*
- c) Abordagens de modelação estruturada, ágil e orientada a objetos*
- d) Responsabilidade Profissional e Social do Engenheiro Informático*

Saber Fazer

- e) Lidar e modelar problemas complexos (Domínio do Problema e da Solução)*
- f) Interpretar, compreender e discutir problemas com requisitos mal definidos (omissões, ambiguidades, contradições, duplicações)*
- g) Usar técnicas de desenvolvimento de software estruturadas, ágeis e orientadas a objetos*
- h) Usar ferramentas adequadas à especificação e desenho*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- a) Software lifecycle, tools, methods and techniques for project management, specification and design of large-scale projects involving large, multidisciplinary teams*
- b) Relevance of project development activities, approaches and models to tackle complex problems in a pragmatic and rigorous way*
- c) Modeling techniques for structured, agile and object-oriented development*
- d) Social, Professional and Ethical responsibility of a Software Engineer*

Application:

- e) Deal with and model complex problems (Problem and Solution)*
- f) Interpret, understand and discuss badly defined requirements (omissions, ambiguities, contradictions, duplications)*
- g) Use techniques for structured, agile and object-oriented software development*
- h) Use adequate tools to support software specification and design*

3.3.5. Conteúdos programáticos:

Parte 1:

Introdução aos Processos de Desenvolvimento de Software

- Utilidade dos processos*
- Modelos (cascata, incremental, prototipagem rápida, ágil) e fases do ciclo de vida*
- Paradigmas de elicitação e modelação mais importantes*

- Estruturado

- Orientado a objetos

- Ágil (Scrum, xProgramming)

Requisitos Funcionais e Não-Funcionais

UML

- Requisitos: Diagramas de Caso de Uso*
- Comportamento: Diagramas de Atividade, Sequência e Estados*
- Estrutura: Diagramas de Classes, Objetos e Pacotes*
- OCL (Object-constraint language)*

Parte 2:

Princípios de Arquitetura e Desenho

- Desenho arquitetural e de detalhe*
- Separação de assuntos*
- Coesão e acoplamento*
- Reutilização de estruturas standard*

Ferramentas e Ambientes

UML

- *Arquitetura: Modelos de Componentes e Instalação*
- *Persistência: Derivar modelos Entidade-Relação*

Parte 3:

Introdução à gestão de projetos

- *Atividades de gestão*
- *Planeamento e Agendamento de projetos*
- *Gestão de riscos*

Código de ética Profissional

3.3.5. Syllabus:

Part 1:

Software Development Processes

- *Usefulness of software development processes*
- *Models and phases (waterfall, incremental, rapid-prototyping, agile) of software lifecycle*

Overview of elicitation and modeling paradigms

- *Structured*
- *Object-Oriented*
- *Agile (Scrum, XProgramming)*

Functional and Non-Functional Requirements

UML

- *Requirements: Use Case Diagrams*
- *Behavior: Activity, Sequence and state diagrams*
- *Structure: Class, Object and Package diagrams*
- *OCL (Object-constraint language)*

Part 2:

Architectural and design principles

- *Architectural design and detailed design*
- *Separation of concerns*
- *Coupling and cohesion*
- *Reuse of standard structures*

Tools and Environments

UML

- *Architecture: Component and Installation Models*
- *Persistency: Deriving Entity-Relationship models*

Part 3:

Introduction to Project Management

- *Management activities*
- *Project Planning and Scheduling*
- *Risk management*

Code of Professional Ethics

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os objetivos da UC alinham pela definição do que é o Engenheiro Informático, enquanto profissional, para atacar sistematicamente problemas complexos com processos escaláveis assentes em reutilização planeada, e enquanto membro de equipas grandes e multidisciplinares. Os conteúdos programáticos baseados no currículo ACM e numa série de cursos reputados em Universidades Internacionais, são apoiados por bibliografia educativa (vide bibliografia sugerida) considerada de referência na área. Estudam-se os ciclos de desenvolvimento, as metodologias e técnicas de gestão (planeamento e risco) e desenvolvimento de projetos, acompanhadas por técnicas de modelação usadas frequentemente na Indústria de Software (como o UML).

Fazendo uma correspondência entre os objetivos apresentados anteriormente e os conteúdos temos:

- a) e b) são respondidos com o conteúdo não UML das Partes 1 e 2, e avaliado com dois testes.*
- c) é respondido com as Partes 1 e 2 relativas ao UML (nomeadamente em ambiente de laboratório), avaliados nas componentes práticas 1 e 2 do projeto.*
- O objetivo d) é respondido com a Parte 3 e avaliado no último componente do projeto.*

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The objectives of this course align well with what should be expected from a Software Engineer, as a professional, by tackling complex problems with scalable processes based on planned reuse, and also as a member of multidisciplinary and large teams. The syllabus is based on the ACM curriculum and on a series of courses in well-respected International Universities, and is supported by educational literature (see suggested bibliography) considered the reference area. This course teaches software development lifecycles, methodologies and techniques for project management (planning and risk) and development, grounded on standard modeling techniques often used in the Software Industry (like UML).

Establishing a correspondence between the course objectives and its syllabus:

a) and b) are introduced in the non-UML Parts 1 and 2, and are evaluated with the written tests.

c) is introduced in the UML Parts of 1 and 2 (exercised in the lab), and evaluated through group project parts 1 and 2.

Goal d) is introduced in Part 3 and is evaluated by the third part of the project.

3.3.7. Metodologias de ensino (avaliação incluída):

Aulas teóricas (duas aulas semanais de 1,5 horas cada) expõem a matéria, ilustrada com exemplos e oradores convidados que relatam a sua experiência na realidade empresarial. As aulas de laboratório (3 horas) serão dedicadas ao uso de ferramentas, por exemplo MagicDraw, e a apresentar o UML com diversas ilustrações e exercícios que os alunos deverão realizar nos laboratórios. Assim, enquanto as aulas teóricas são de âmbito largo nos tópicos que abordam, as aulas práticas concentrar-se-á fundamentalmente no UML, com algumas ligações a técnicas concretas de outros paradigmas.

A avaliação é calculada pela média ponderada de: um projeto realizado em grupos e dividido em 3 partes (P1 20%, P2 15% e P3 15%); e dois testes, cada um com um peso de 25%. Os alunos que não satisfizerem qualquer uma das partes acima são classificados como "Reprovados".

O exame de recurso é reservado para os alunos com desempenho satisfatório, cuja nota final ponderada totalize pelo menos 9,5 pontos (em 20).

3.3.7. Teaching methodologies (including evaluation):

Two weekly lectures of 1,5 hours each teach the theoretical concepts and one lab class of 3 hours focuses on application. While the lectures are dedicated to expose the various topics illustrated with examples and followed by talks by industry experts reporting their experience, the lab is dedicated to the use of tools, like MagicDraw, and to present UML, followed by exercises.

The evaluation is composed of a Project (with three distinct phases P1 20%, P2 20% and P3 10% of the total grade) developed in teams, and two individual tests (each counting 25% of the total grade). All these evaluation components are mandatory. Students failing to satisfy this are classified with "Reprovado" (fail).

Access to the ressit exam is conditional, and only allowed to students with a final grade (computed as the weighted sum of all evaluation components) of at least 9,5 out of 20.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

O objetivo principal da cadeira é apresentar técnicas sistemáticas e escaláveis para abordar o desenvolvimento de um projeto real e complexo. Para tal, serão dados exemplos inspirados em situações reais, para além de se convidar oradores com relevante experiência profissional para contribuir com pequenas palestras relatando a realidade no ambiente empresarial. É neste contexto que o projeto a ser realizado em grupo de alunos, como forma de avaliação, assume um papel fundamental de concretização das técnicas e métodos apresentados nas aulas teóricas. As aulas práticas, realizadas num laboratório com o software adequado, possibilitam a familiarização do aluno com ferramentas de apoio ao planeamento, análise e desenho de software, e permitem-lhe o desenvolvimento (em UML) de um projeto complexo e de outros exercícios relacionados com fases específicas do projeto. (O mapeamento desta metodologia de ensino com os objetivos já foram discutidos em secções anteriores.)

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The main goal of the course is to present systematic and scalable techniques and methods to deal with the software development process of real projects, complex by nature. Examples inspired in real life will be presented, and experts from industry will be invited to report on their experience in real projects. The students will engage in a group project along the semester, therefore supporting the instantiation of the theoretical topics taught along the semester. The labs are equipped with tools (e.g., CASE tools) that support the project planning and the corresponding system's analysis and design phases (in UML), as well as the development of a common case study and the practice with some exercises related to specific phases of the case study. (The mapping of the methodologies with the goals were already discussed in the previous sections.)

3.3.9. Bibliografia principal:

Software Engineering: A Practitioner's Approach, R. Pressman, McGraw-Hill, 2009
The Unified Modeling Language User Guide, G. Booch, J. Rumbaugh & I. Jacobson, Addison-Wesley; 2nd Ed, 2005
UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition), Addison-Wesley; 2nd Ed, 2005
Writing Effective Use Cases, A. Cockburn, Addison Wesley, 2000
The Object Constraint Language: Getting Your Models Ready for MDA (2nd Ed.), A. Kleppe, J. Warmer, Addison-Wesley, 2003
The Unified Software Development Process, I. Jacobson, G. Booch & J. Rumbaugh, Addison-Wesley, 1999
A code of ethics and professional practice for software engineering, ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices, 2004
Extreme Programming Explained: Embrace Change, 2nd Edition, K. Beck, C. Andres, Addison-Wesley, 2004
Agile Software Development with Scrum, K. Schwaber, M. Beedle, Prentice-Hall, 2001
Modern Structured Analysis, Edward Yourdon, Prentice Hall, 1988

Mapa IV - Redes de Computadores / Computer Networks**3.3.1. Unidade curricular:**

Redes de Computadores / Computer Networks

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Augusto Legatheaux Martins (Responsável e Regente) (T-84h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Henrique João Domingos (P-84h), Sérgio Marco Duarte (P-84h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Esta é uma unidade curricular obrigatória que introduz o estudante aos fundamentos das redes de computadores.

Saber

- *Redes de computadores: terminologia, organização, serviços, componentes e princípios estruturantes*
- *Estruturação de aplicações e protocolos aplicativos: requisitos das aplicações e modelos de comunicação e coordenação*
- *Noções fundamentais sobre canais de dados e redes comutadas*
- *Protocolos de transporte: requisitos e técnicas específicas*
- *Protocolos de transporte seguros*
- *Encaminhamento: princípios, endereçamento e algoritmos mais comuns*
- *Técnicas de controlo de acesso em canais partilhados*

Saber Fazer

- *Analisar um protocolo e perceber o seu funcionamento*
- *Conhecer e saber utilizar interfaces de comunicação*
- *Analisar, estruturar e desenvolver pequenas aplicações distribuídas*
- *Modelização de um protocolo de transporte e modelização de uma rede*
- *Simulação de protocolos*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This is a mandatory first course on fundamentals of computer networks..

Knowledge:

- *Introduction to computer networks: terminology, organization, services, components and structuring principles*
- *Application and protocols: applications requirements and models of communication and coordination*
- *Fundamentals of communication channels and switched networks*
- *Transport protocols: specifications and implementation techniques*
- *Security at the transport level*
- *Routing: principles, addressing and algorithms*
- *Data-link layer issues of shared wired and wireless channels*

Application

- *Analyze a protocol and understand its details*
- *Usage of communication interfaces and their usage*
- *Structure and implement small distributed applications*
- *Transport protocols and network analysis and modeling*
- *Protocol simulation*

3.3.5. Conteúdos programáticos:

Conteúdos programáticos

- *Redes de computadores: terminologia, organização, serviços, componentes e princípios estruturantes*
- *Noções fundamentais sobre canais de dados e redes comutadas*
- *Estruturação de aplicações e protocolos aplicativos: requisitos das aplicações e modelos de comunicação e coordenação*
- *Protocolos de transporte: requisitos e técnicas específicas*
- *Protocolos de transporte seguros – breve introdução*
- *Encaminhamento: princípios, endereçamento e algoritmos mais comuns*
- *Técnicas de controlo de acesso em canais partilhados*

3.3.5. Syllabus:

Syllabus

1. *Introduction to computer networks: terminology, organization, services, components and structuring principles*
2. *Application and protocols: applications requirements and models of communication and coordination*
3. *Fundamentals of communication channels and switched networks*
4. *Transport protocols: specifications and implementation techniques*
5. *Security at the transport level - a short introduction*
6. *Routing: principles, addressing and algorithms*
7. *Data-link layer issues of shared wired and wireless channels.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A organização e forma de apresentação dos tópicos é especialmente pensada para alunos de Engenharia Informática que devem ter uma visão algorítmica e computacional das redes de computadores. A aptidão para o desenvolvimento de aplicações distribuídas, usando as interfaces de rede de nível transporte, devem ser exercitadas. A sequência de apresentação dos temas é parcialmente top-down. Aspectos fundamentais como a estruturação e modelização das redes, as noções fundamentais sobre as interfaces de rede, a estruturação das aplicações e seus protocolos, o desenho e análise dos protocolos de transporte fiáveis, o encaminhamento e sua análise, e o funcionamento dos canais partilhados são os temas de base que se procura que o aluno domine. Estes aspectos são de forma geral reconhecidos como os aspectos fundamentais das redes de computadores que os alunos de Engenharia Informática devem dominar. Dada a sua relevância, são fornecidos também conhecimentos de base sobre canais seguros (especialmente SSL).

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The selected topics as well as their sequence is specially fitted to Computer Science students with a bias towards an algorithmic and computational view of computer networks. Their ability to develop small distributed applications using transport-level network interface functions is exercised. The topics sequence is partially top-down.

Fundamental issues like modelling and structuring of computer networks, the main abstractions related to transport interfaces, applications structure and protocols, design and analysis of reliable transport protocols, routing and its analysis, and shared channels and medium control access, are recognized as the main computer network related topics for Computer Science students. Given its relevance, base knowledge on secure channels is also introduced (specially SSL).

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas expõe-se e discute-se o programa da cadeira. Em algumas aulas teóricas resolvem-se e discutem-se problemas tipo que envolvem a modelização e a análise de um protocolo ou de uma situação. Nas aulas práticas são desenvolvidos diversos trabalhos cujo objetivo é levar o aluno a realizar trabalhos experimentais sobre os tópicos mais importantes da matéria. Para além de cumprirem os objetivos dos trabalhos, os alunos devem depois relacionar os resultados obtidos com os tópicos da matéria que são relevantes. Em várias aulas procede-se à resolução de problemas tipo e também à análise de protocolos.

Componentes da avaliação

- *Dois testes abrangendo a teoria e a prática envolvidas na unidade curricular.*

- *Realização de vários testes rápidos e de fichas de trabalhos de pequena dimensão sujeitos a avaliação quantitativa e qualitativa.*
- *Um ou dois mini projetos de programação de aplicações e protocolos.*

3.3.7. Teaching methodologies (including evaluation):

Lectures are intended to support the instructor's presentation of fundamental issues, as well as discussion of such topics with students. In some lectures, standard problems are solved and discussed and protocols are analyzed.

Laboratory sessions take place in a computer laboratory and their aim is to solve problems and learn how to build network applications and protocols. These exercises, protocols and applications use the principles presented during lectures.

Assessment components

- *Two mid-term tests encompassing theory and methods of the curricular unit.*
- *Several quizzes and small assignments.*
- *One or two small sized programming projects.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As metodologias de ensino são tradicionais e envolvem: exposição dos aspetos essenciais da matéria, resolução de exercícios com modelização e análise de aplicações, configurações e protocolos, realização de trabalhos práticos no laboratório e projetos de programação.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Teaching methodologies are well known and established and encompass: presentation and discussion of the main subjects, problem solving requiring applications and protocols modelling and analysis, completion of small laboratory exercises and programming projects.

3.3.9. Bibliografia principal:

Bibliografia de base

- *James F. Kurose and Keith W. Ross, "Computer Networking - A Top-Down Approach Featuring the Internet," Addison Wesley — Pearson Education, 2012, ISBN-10: 0136079679, 6th Edition*
- *Illiotte Rusty Harold, "Java Network Programming," O Reilly & Associates, 3rd Edition, 2004, ISBN 0-596-00721-3, 410 pages*

Bibliografia complementar

- *A. Tanenbaum, "Computer Networks," Prentice-Hall, 5th Edition, 2010*
- *Kenneth L. Calvert and Michael Donahoo, "TCP/IP Sockets in Java: Practical Guide for Programmers," Morgan Kaufman, 2002, ISBN 1-55860-685-8, 72 pages*
- *W. Richard Stevens, "TCP/IP Illustrated - Vol. 1 - The Protocols," Addison-Wesley, 1995*

Mapa IV - Atividade Prática de Desenvolvimento Curricular/ Practical Activity for Curricular Development

3.3.1. Unidade curricular:

Atividade Prática de Desenvolvimento Curricular/ Practical Activity for Curricular Development

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Abílio Duarte de Medeiros (S-14h; OT-30h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Maria Armada Simenta Rodrigues Grueau (P-28h)

João Ricardo Viegas da Costa Seco (P-28h)

João Carlos Gomes Moura Pires (P-28h; OT-20h)

João Manuel dos Santos Lourenço (P-56h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Oferecida na transição entre o blocos base e consolidação esta UC concede aos estudantes experiência

integrada, em contexto aberto (fora da sala de aulas), de atividade técnica aplicada, adequada ao seu nível de conhecimentos. O contexto variável, por opção dos estudantes, poderá ser equipa de investigação, laboratório, ambiente empresarial / industrial. Esta UC complementa as UCs PIPP e PIIC do “Perfil Curricular” da FCTUNL da forma explicada abaixo.

Saber: Interpretar os requisitos e contexto duma situação problemática de alguma dimensão em Ciência e Engenharia Informática. Desenvolver, de forma assistida mas autonomizante, soluções, técnicas e métodos a utilizar.

Saber Fazer: Implementar um ou mais componentes necessários para a solução, usando instrumentos e ferramentas de desenvolvimento industriais. Validar, depurar e avaliar soluções obtidas.

Competências Transversais: Reforçar visão sistema, autonomia, capacidade crítica e trabalho em equipa.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Offered in the interface between the core tier and the consolidation tier, this CU provides students with an integrated experience of a technical focused hands-on activity, adequate to their knowledge level, and in open, out of classroom, context. The environment may vary, as chosen by the students, e.g. a research team, a lab, an internship in industry, in complement to the offer UROP and UPOP included in FCT-UNL “Curricular Profile” (see below).

Knowledge: Understand the requirements and context of a problematic situation of a certain dimension in the field of Computer Science and Engineering. Develop convenient solutions, under a degree of supervision promoting autonomy. Identify the techniques and methods to use.

Application: Implement one or several components needed for the solution, using industrial strength development tools. Validate, debug, and evaluated the solutions delivered.

Transferable Skills: Enforce system view, autonomy, critical thinking, and teamwork abilities.

3.3.5. Conteúdos programáticos:

O programa de uma unidade curricular com estas características não segue os termos habituais, sendo no entanto de salientar os tópicos seguintes: Familiarização com os conceitos, as tecnologias e as metodologias necessárias ao longo da atividade. Integração prática de conhecimentos de base adquiridos em várias unidades curriculares de carácter mais fundamental no bloco base, nas várias áreas científicas do curso.

Familiarização com o contexto humano, social e técnico em que o trabalho se desenvolve. Técnicas de gestão de projeto e de trabalho em equipa. Aspectos fundamentais de práticas de qualidade. Aspectos de segurança informática. Aspectos fundamentais de usabilidade de sistemas informáticos. Técnicas de comunicação (documentação, apresentação, posters).

3.3.5. Syllabus:

The syllabus of a curricular unit with these characteristics does not follow the usual terms, however the following topics will be covered: Familiarization with the concepts, technologies and methodologies required for tackling the chosen activity. Integration through practical hands-on experience of the basic knowledge obtained in several fundamental curricular units of the core tier, in the several scientific areas of the program.

Understanding of the human, social, and technical context in which the activity takes place. Project management methodologies and of teamwork. Principles of quality management. Aspects of information security. Aspects of usability of informatic systems. Communication skills (documentation, presentation and posters).

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos programáticos endereçam precisamente as competências necessárias ao progresso dos estudantes no sentido de obterem as competências enunciadas. Os projetos e atividades a desenvolver pelos estudantes são desenhados e selecionados pela equipa docente de forma a garantir que aqueles disponham de várias oportunidades de atingirem os vários objetivos elencados nos conteúdos programáticos. Como exemplos possíveis de projetos / atividades a realizar pelos estudantes incluem-se atividades como: o desenvolvimento de uma aplicação web para um dispositivo móvel, incluindo backend e frontend (projeto interno), análise experimental de performance de uma infraestrutura de comunicação (atividade de introdução à investigação), o desenvolvimento de uma extensão a um sistema de informação empresarial (projeto externo em empresa).

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus addresses the topics necessary to the good progress of students towards the acquisition of the intended competencies. The projects and activities to be developed by the students are designed and selected by the teaching team so to ensure that the former will be exposed to appropriate opportunities for developing the

said objectives. As examples of possible projects / activities to be developed by students we may consider: development of a web application for mobile devices, including backend and frontend (internal or external project), experimental validation of the performance of a network infrastructure (introductory research), or the development of an extension to an corporate information system (external project in a firm).

3.3.7. Metodologias de ensino (avaliação incluída):

Esta unidade curricular visa o desenvolvimento de uma atividade coerente, realizada no Departamento ou no exterior, e envolve a integração de competências e conhecimentos endereçadas de forma independente nas várias UCs do bloco base, nomeadamente programação, modelação e desenho de software, bases de dados, algoritmos e estruturas de dados, linguagens de programação, sistemas e redes de computadores, interfaces pessoa-máquina, e outras, dependendo de cada projeto / atividade. O ensino segue os seguintes métodos: apoio tutorial, aulas tutoriais, sessões de uniformização (por vezes envolvendo convidados exteriores), e monitorização intermédia. No fim da unidade os estudantes devem apresentar publicamente e através de relatório os resultados do seu trabalho. É importante garantir que os parâmetros e métodos de avaliação permitem avaliar de forma correta em que medida o estudante atingiu os objetivos de aprendizagem, independentemente do contexto em que o trabalho seja realizado.

3.3.7. Teaching methodologies (including evaluation):

This curricular unit targets the development by the students of a coherent activity, mostly out of the controlled class room context, to be performed either in the Department or outside, and involving integration of competencies and knowledge addressed independently in several curricular units of the core tier, namely, programming and software design and development, computer systems and networks, human machine interaction, among others, depending on the project / activity at hand. The teaching approach involves essentially tutorial support and uniformisation sessions (sometimes by external guests), and several progress milestones. By the end of the curricular unit the student (team) must publicly present by means of a written report the results of his work. A key aspect to be taken in consideration is the uniformity of parameters and evaluation methods to evaluate to what extent the student has reached the learning outcomes, independently of the working environment context.

3.3.8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular:

Da descrição realizada acima, tendo em conta a natureza especial desta unidade curricular, decorre que existe uma adequação correta entre as metodologias de ensino e os objetivos de aprendizagem da unidade curricular. Como exemplos possíveis de projetos / atividades a realizar pelos estudantes incluem-se atividades como: o desenvolvimento de uma aplicação web para um dispositivo móvel, incluindo backend e frontend (projeto interno), análise experimental de performance de uma infraestrutura de comunicação (atividade de introdução à investigação), o desenvolvimento de uma extensão a um sistema de informação empresarial (projeto externo em empresa). A avaliação tem em conta os seguintes aspetos: 1) O desempenho e participação do aluno nas sessões tutoriais e, de um modo geral, na interação com o(s) docente(s) responsável ao longo da UC. Esta avaliação faz-se por observação e indicadores registados de desempenho em relação a presença, controlo de evolução do trabalho, pró-atividade e autonomia, controlo e acompanhamento do projeto bem como do nível de "reporting/tracking" das tarefas do projeto. 2) O trabalho desenvolvido / qualidade e extensão do projeto. Avalia a qualidade, extensão e cobertura de requisitos do projeto, (com avaliação final de objetivos alcançados, completude face a especificações, criatividade, robustez e usabilidade e aspetos de qualidade de produto e processo), incluindo a qualidade documentação técnica a entregar. 3) A avaliação final consiste sempre numa discussão pública com o(s) autor(es) do trabalho, tendo em conta a opinião das entidades envolvidas no contexto de acolhimento (equipa de investigação, coordenadores em empresa, etc). Acresce que unidades curriculares com aspetos semelhantes à que aqui propomos, mesmo que com diferenças na sua estratégia de implementação, visando o envolvimento supervisionado dos estudantes em atividades de realização prática, numa fase intermédia da formação, são adotadas com sucesso em várias departamentos de referência a nível internacional, como por exemplo o Imperial College London, Cambridge Computer Lab, e outros. No contexto do curso, esta UC complementa as UCs PIPP e PIIC do "Perfil Curricular" da FCTUNL de forma flexível / complementar, permitindo por exemplo ao estudante uma experiência de introdução à investigação no PIIC e uma experiência de atividade em empresa na APDC, ou uma experiência de atividade PIPP e uma experiência mais intensa de introdução à investigação ou projeto na APDC, entre outras combinações possíveis.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Results from the description given above, and considering the special nature of this curricular unit, that there is a correct coherence between the teaching methodologies and the learning outcomes.

As examples of possible projects / activities to be developed by students we may consider: development of a web application for mobile devices, including backend and frontend (internal or external project), experimental

validation of the performance of a network infrastructure (introductory research), or the development of an extension to a corporate information system (external project in a firm). The evaluation takes into account the following aspects: 1) The performance and participation of the student in the tutorial sessions, and, in general, his interaction with the teaching staff. This evaluation is performed by individual observation, and by collecting performance indicators relative to attendance, work evolution control, proactivity and autonomy, project follow up, and reporting / tracking of the several tasks as well. 2) The developed work and its extension and quality. This item evaluates the overall quality, extension, and requirements coverage, including the final evaluation of objectives achieved, creativity, completeness with respect to the initial specs, robustness, usability, and process quality. 3) The final evaluation includes a public discussion with the author(s) of the work and the teaching staff, taking into consideration the opinion of the various involved participants (research team, coordinating hosts in external firm, etc). Additionally, it should be mentioned that curricular units with similar goals to the one here proposed, even if with differences in implementation strategies, targeting the supervised involvement of students in hands-on activity in an intermediate phase of the educational program, have been adopted with success in several reference international CS departments, such as the Imperial College London, the Cambridge Computer Laboratory, and others. In the context of the program, this CU complements the UROP and UPOP CUs common to the FCTUNL "Curricular Profile" in a flexible / complementary way, allowing for instance a student to develop some introduction to research in the UROP unit and an experience of activity in a company in PACD, or an UPOP experience and a more intensive involvement in research or project in PACS, among other possible combinations.

3.3.9. Bibliografia principal:

Depende de cada atividade / projeto. São recomendados aos estudantes várias referências sobre a área concreta do projeto / atividade;

Depending on the activity / project. Several references are recommended to the students, appropriate to each concrete project / activity.

Mapa IV - Programa de Introdução à Prática Profissional / Undergraduate Practice Opportunities Program

3.3.1. Unidade curricular:

Programa de Introdução à Prática Profissional / Undergraduate Practice Opportunities Program

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (OT-7h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Ricardo Viegas da Costa Seco (Regente) (OT-7h)

Todos os docentes da área científica principal do curso de Mestrado Integrado em Engenharia Informática (OT-7h) / All professors in the main scientific area of the Integrated Master in Computer Science and Engineering (7h).

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

O Programa de Introdução à Prática Profissional (PIPP) visa promover a participação de estudantes, desde cedo na sua formação académica, em atividades em ambiente empresarial.

Através do programa, o estudante que dele participe terá contacto com trabalhos de engenharia, no dia a dia, numa empresa. Tomará conhecimento do modo de funcionamento de projetos de engenharia em ambiente empresarial. Desenvolverá competências transversais de trabalho em grupo, de comunicação escrita e oral, e aprendizagem em autonomia.

Deverá ainda adquirir conhecimentos e, eventualmente, aptidões técnicas específicas na área concreta do trabalho que o estudante desenvolve na empresa.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The Undergraduate Practice Opportunities Program (UPOP) program aims at promoting the participation of students, since early in their academic career, in practical activities in non-academic environment.

Through UPOPs, the student will have contact with the daily activities of engineering projects in a company. By this contact, the student gets to know how engineering projects develop, in practice. (S)he is expected to develop transferable skills in working in teams, oral and written communication, and independent learning.

Depending on the specific work developed by the student in the company, (s)he will acquire specific knowledge on the subject area and, possibly, also some specific technical skills relevant to the placement.

3.3.5. Conteúdos programáticos:

Os conteúdos programáticos específicos dependem do projeto concreto escolhido pelo estudante no programa

3.3.5. Syllabus:

The concrete syllabus depends on the specific project chosen by the student in the program.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Não aplicável.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Not applicable

3.3.7. Metodologias de ensino (avaliação incluída):

A comissão científica do curso mantém uma lista de ofertas de participação em projetos em ambiente empresarial, no âmbito do PIPP. Cada entrada na lista apresenta o nome da empresa, o projeto em que é enquadrado, um plano de trabalhos sumário, o período em que as atividades são desenvolvidas, e os orientadores na empresa e científico.

O estudante escolhe um dos projetos da lista. Havendo vários interessados numa mesma participação, cabe ao orientador na empresa escolher o estudante a participar.

O estudante cumpre o plano de trabalho com orientação tutorial, no período designado, devendo esse período, em princípio, coincidir com o período entre o final dos exames e o início do semestre seguinte. As atividades de PIPP podem também ser consideradas como parte de estágios mais alargados (e.g. estágios de Verão). A avaliação é feita por relatório onde o estudante descreve as atividades desenvolvidas, podendo ser complementada com informação dos orientadores colhida no decurso do trabalho.

3.3.7. Teaching methodologies (including evaluation):

The scientific committee of the study cycle keeps a list of UPOP offers, for the participation of students in projects in non-academic environment. Each entry in the list has the name of the company, the project in which the student will be integrated, the work plan, the period in which the activities take place, and the names of the supervisor in the company and the scientific supervisor.

The student chooses one of the UPOP's offers. If several choose the same offer, it is up to the supervisor in the company to select the student. The student carries out the work plan with supervision, in the designated period, which in principles is the period between the end of exams and the beginning of the next semester. UPOP projects can also be considered as part of larger internships in a company (e.g. summer internships). The assessment is made by a final report, where the student describes the activities, and can be complemented with information collected by the supervisors during the period.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A formação da lista de ofertas no âmbito no Programa de Introdução à Prática Profissional, selecionadas pela comissão científica do curso permitirá de facto, aos estudantes interessados em seguir este programa, a participação em atividades em ambiente empresarial.

Através dessa seleção, é garantido que as atividades do estudante, supervisionadas pelo orientador, são integradas em equipas na empresa. Do contacto com a equipa e orientador na empresa, que durante o período das atividades será praticamente diário, resulta necessariamente um contacto e conhecimento das práticas de trabalho de engenharia da empresa. Se o trabalho exigir conhecimentos e/ou aptidões específicas estas terão que ser adquiridas pelo estudante, em autonomia, embora com orientação do orientador científico.

As técnicas de comunicação são exigidas, e testadas, para a avaliação final.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The UPOP offers, selected by the scientific committee, allow interested students to participate in real projects carried out in non-academic environment.

Through the selection process it is guaranteed that the activities of the student will be integrated in teams in the company. From the contact with the team, and with the supervisor in the company, which will be daily or close to daily, the student will get to know the work practices of the company in engineering projects. If the work plan requires specific knowledge and technical skills, these are to be acquired by the student with independent learning, with supervision from the academic supervisor.

The communication skills are required, and assessed, in the final evaluation.

3.3.9. Bibliografia principal:

Depende do projeto específico escolhido por cada estudante.

Depends on the specific project chosen by the student.

Mapa IV - Programa de Introdução à Investigação Científica / Undergraduate Research Opportunities Program

3.3.1. Unidade curricular:

Programa de Introdução à Investigação Científica / Undergraduate Research Opportunities Program

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (OT-7h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Ludwig Krippahl (Regente) (OT-7h)

Todos os docentes da área científica principal do curso de Mestrado Integrado em Engenharia Informática (OT-7h) / All professors in the main scientific area of the Integrated Master in Computer Science and Engineering (7h).

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

O Programa de Introdução à Investigação Científica (PIIC) visa promover a participação de estudantes, desde cedo na sua formação, em projetos de investigação científica coordenados por docentes e investigadores da faculdade.

Através do programa, o estudante que dele participe deverá ter contacto com práticas de investigação científica e adquirir conhecimento do modo de funcionamento de projetos de investigação. Desenvolverá aptidões de apresentação e explicação de resultados científicos, e competências transversais de trabalho em grupo, de comunicação escrita e oral, e aprendizagem em autonomia.

Deverá ainda adquirir conhecimentos e, eventualmente, aptidões técnicas específicas na área concreta do projeto em que o estudante esteja envolvido.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The Undergraduate Research Opportunities Program (UROP) program aims at promoting the participation of students, since early in their academic career, in research projects developed by academic staff of the faculty. Through UROPs, the student will have contact with scientific research environment and gain knowledge of how research projects work. The student will develop skills in presenting and explaining research results, and transferable skills of working in teams, oral and written communication, and independent learning.

Depending on the specific project chosen by the student, (s)he will acquire specific knowledge on the subject area and, possibly, also some specific technical skills in the project area.

3.3.5. Conteúdos programáticos:

Os conteúdos programáticos específicos dependem do projeto concreto escolhido pelo estudante no programa.

3.3.5. Syllabus:

The concrete syllabus depends on the specific project chosen by the student in the program.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Não aplicável.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Not applicable.

3.3.7. Metodologias de ensino (avaliação incluída):

A comissão científica do curso mantém uma lista de ofertas de participação de estudantes em projetos de investigação, no âmbito do Programa de Introdução à Investigação Científica. Cada entrada nessa lista deverá apresentar o projeto em que o estudante será enquadrado, um plano de trabalhos sumário, e o orientador científico.

O estudante escolhe a participação num dos projetos da lista. Havendo vários estudantes interessados numa mesma participação, cabe ao orientador científico escolher o estudante a participar.

O estudante cumpre o plano de trabalho ao longo do semestre, com especial incidência no período entre o final da época de exames e o início do semestre seguinte, tendo durante esse período orientação tutorial.

A avaliação é feita por relatório final das atividades desenvolvidas, podendo ser complementada com informação do orientador, de avaliação contínua que este tenha feito do trabalho ao longo do semestre.

3.3.7. Teaching methodologies (including evaluation):

The scientific committee of the study cycle keeps a list of UROP offers, for the participation of students in research projects. Each entry in the list must present the research project in which the student will be integrated, the work plan for the student, and the name of the scientific supervisor.

The student chooses one of the UROP's offers. If several students choose the same offer, it is up to the supervisor to select one of the students.

The student carries out the work plan along the semester, with special incidence in the period between the end of exams and the beginning of the next semester.

The assessment is made by a final report, describing the activities and results obtained. The assessment can be complemented with further information collected by the supervisor during the activities.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A existência de uma oferta atualizada de participação em projetos de investigação científica permitirá de facto, aos estudantes interessados em seguir este programa, a participação em atividades de investigação.

Sendo esta oferta sempre, necessariamente, integrada em projetos de investigação em curso na faculdade, sob a coordenação de docentes ou investigadores, projetos esses que envolvem equipas de investigação, é oferecida ao estudante a oportunidade de trabalho em equipa. Do contacto com a equipa de investigação, que durante o período intercalar (entre o final da época de exames e o início do semestre seguinte) será praticamente diário, resulta necessariamente um contacto e conhecimento das práticas de investigação da equipa. Se o trabalho exigir conhecimentos e/ou aptidões específicas essas terão que ser adquiridas pelo estudante, em autonomia, embora com orientação do docente orientador.

As técnicas de comunicação são exigidas, e testadas, para a avaliação final.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The existence of an up-to-date list of UROP offers allows interested students to participate in real research activities carried out by academic staff of the Faculty.

Given that the offer must be integrated in ongoing research projects, carried out by teams of researchers, it is guaranteed that the student will work in a team, and necessarily given the opportunity to develop skills of team work. From the contact with the research team, which during the intercalary period (between the end of exams and the beginning of the next semester) will be daily or close to daily, the student will get to know scientific research practices of the project. If the work plan requires specific knowledge and technical skills, these are to be acquired by the student in independent learning, with supervision.

The communication skills are required, and assessed, in the final evaluation.

3.3.9. Bibliografia principal:

Depende do projeto específico escolhido por cada estudante.

Depends on the specific project chosen by the student.

Mapa IV - Introdução à Investigação Operacional / Introduction to Operational Research

3.3.1. Unidade curricular:

Introdução à Investigação Operacional / Introduction to Operational Research

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Ruy Araújo da Costa (Responsável e Regente) (T – 28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Maria do Carmo Proença Caseiro Brás (P – 84h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

No final desta unidade curricular o estudante terá adquirido conhecimentos, aptidões e competências que lhe permitam:

- formular e resolver um problema de Programação Linear, incluindo utilizar o Algoritmo Simplex Revisto e conhecer a Análise de Sensibilidade e de Pós-Optimalidade;*
- formular e resolver resolver um problema de Programação Linear Inteira utilizando o Algoritmo Branch and Bound;*
- formular e resolver resolver um Problema de Transportes;*
- analisar problemas de Decisão, em contextos de incerteza ou de risco , bem como representar a Árvore de Decisão de um problema de decisões sequenciais, fazendo a sua análise;*
- dominar os modelos básicos de Filas de Espera, incluindo os modelos com prioridades, os modelos envolvendo distribuições não exponenciais e as redes de de filas de espera;*
- conhecer técnicas de geração de números pseudo-aleatórios, construir fluxogramas correspondentes a rotinas e modelos de Simulação.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

After this curricular unit, a student will be able to:

- formulate and solve a Linear Programming problem, including the revised form of the Simplex algorithm, carry out sensitivity analysis;*
- formulate and solve an Integer Linear Programming problem, using the Branch and Bound algorithm;*
- formulate and solve a Transportation Problem;*
- analyze Decision making problems (decision in uncertainty and risk situations); draw and analyze the decision tree in sequential decision making problems;*
- master the basic queueing models, including priority-discipline models, models involving non-exponential distributions and queueing networks;*
- master pseudo-random numbers generation techniques; draw flowcharts of Simulation models.*

3.3.5. Conteúdos programáticos:

1 – Programação Linear

1.1– Formulação de Problemas de Programação Linear.

1.2– Algoritmo Simplex.

1.3– Formulação matricial do Simplex. Algoritmo Simplex Revisto.

1.4– Análise de Sensibilidade e de Pós-Optimalidade.

1.5– Programação Linear Inteira: Algoritmo de Branch and Bound.

1.6– Problema e Algoritmos dos Transportes.

2 – Teoria da Decisão

2.1 – Critérios de decisão em Situação de Incerteza e em Situação de Risco.

2.2 – Decisões Sequenciais.

3 – Filas de Espera

3.1 – Modelos de Filas de Espera baseados no Processo de Nascimento e Morte.

3.2 – Modelos envolvendo distribuições não exponenciais.

3.3 – Modelos de Filas de Espera com Disciplina Prioritária.

3.4 – Redes de Filas de Espera.

4 – Simulação

4.1 – Aspectos ""Técnicos"" da Simulação.

4.2 – Geração de Números Pseudo-Aleatórios.

4.3 – Aplicações à simulação de filas de espera.

3.3.5. Syllabus:

1 – Programação Linear

1.1 – Formulating Linear Programming problems.

1.2 – Simplex Algorithm.

1.3 – The Revised Simplex Method.

1.4 – Sensitivity Analysis.

1.5 – *Linear Integer Programming: Branch and Bound Algorithm.*

1.6 – *Transportation Problem and Algorithm.*

2 – *Decision Making Analysis*

3.1 – *Decisions under risk and under uncertainty.*

3.2 – *Sequential decisions.*

3 – *Queueing models*

3.1 – *Queueing models based on the birth-death process.*

3.2 – *Queueing models involving non-exponential distributions.*

3.3 – *Priority-discipline queueing models com Disciplina Prioritária.*

3.4 – *Queueing networks.*

4 – *Simulation*

4.1 – *Generating random numbers: mixed congruential method, inverse transformation, acceptance-rejection method.*

4.2 – *Duration of the simulation/Results precision.*

4.3 – *Application to queueing systems.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos programáticos correspondem à abordagem clássica dos temas correspondentes, feita nomeadamente na Bibliografia de referência da área, p.ex., Hillier e Lieberman, assegurando uma aprendizagem coerente com os objectivos definidos.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus follows important references in the area, namely, Hillier and Lieberman's "Introduction to Operations Research", thus ensuring its coherence with the curricular unit's objectives.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas são apresentados conceitos fundamentais, que são aplicados nas aulas práticas, que decorrem em laboratório computacional.

Há um Texto de Apoio e uma colectânea de Exercícios disponibilizados na plataforma de ensino à distância moodle. Semanalmente, há um teste no moodle, que se auto-corrige e dá feedback aos alunos, ajudando-os na sua aprendizagem.

Os alunos são encorajados a utilizar regularmente estes testes. Se o fizerem e se assistirem regularmente às aulas práticas, podem realizar 4 Testes durante o semestre – avaliação de época normal. Caso contrário, só são avaliados em exame de época de recurso.

Os docentes disponibilizam Horários de Atendimento.

As provas de avaliação são, em geral, de natureza prática, com questões de diferentes graus de dificuldade. Classificações superiores a 17 só são atribuídas após a submissão de um trabalho complementar e sua discussão, assegurando assim que as classificações mais elevadas só são atribuídas a alunos invulgares.

3.3.7. Teaching methodologies (including evaluation):

Fundamentals are introduced in theoretical lessons. In practical lessons, which take place at a computer-lab, students will solve problems, using the fundamentals.

There is a textbook and an Exercise Book, both available in moodle e-learning platform.

Weekly there is a moodle test (self-correcting and with feedback to help the learning process).

Students are encouraged to use regularly these moodle tests and if they do so and attend practical lessons regularly they are allowed to take four tests during the semester; otherwise, they will have to take the full exam.

Teachers are available throughout the semester to help students in need.

Evaluation tests or exams are mainly practical, although with different difficulty levels.

Students with more than 17, should submit an additional project, and discuss it, thus ensuring that top levels are only attributed to outstanding students.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As metodologias de ensino asseguram a transmissão cuidada dos fundamentos teóricos, antes de se iniciar a fase de aplicação. Por outro lado, nas aulas práticas é feita uma aplicação diversificada dos referidos

fundamentos teóricos.

São disponibilizados um texto de apoio, uma colectânea de exercícios e 14 actividades semanais tipo teste que se auto-corrige no moodle, sendo os alunos fortemente encorajados a utilizar essas atividades. Todos os dias, há docentes disponíveis para esclarecer presencialmente as dúvidas dos estudantes. Assim, o processo de aprendizagem é robusto, fortemente assistido, assegurando aos estudantes interessados uma aprendizagem coerente com os objectivos de aprendizagem definidos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Teaching methodologies ensure a careful transmission of the fundamentals, before students start to apply those concepts. In practical lessons, students will apply the fundamentals to solve a great variety of problems. A text book, an exercise book and 14 weekly moodle-tests (self correcting and with feedback to each answer) are available and students are strongly encouraged to use these tests. Everyday there is at least one teacher available to help students with their difficulties in his/her office. Therefore, the learning process is robust, highly guided, ensuring interested students a smooth learning coherent with the defined learning outcomes.

3.3.9. Bibliografia principal:

- 1 – “Investigação Operacional” (1996), Valadares Tavares et al - Mc Graw Hill
- 2 – “Introduction to Operations Research” (1990 - 5ªEd.), Hillier, Lieberman - Mc Graw
- 3 – “Operations Research - An Introduction”(1992 - 5ª Ed.) Taha - Prentice Hall
- 4 – “Programação Linear” (Vol. 1) (1984), Ramalhete et al - Mc Graw Hill
- 5 – “Elementos de apoio às aulas de Introdução à Investigação Operacional”, “Enunciados de Exercícios de Introdução à Investigação Operacional”, Ruy A. Costa (text book and exercise book - pdf files available in moodle)

Mapa IV - Aspetos Socio-Profissionais da Informática / Social and Professional Aspects of Informatics

3.3.1. Unidade curricular:

Aspetos Socio-Profissionais da Informática / Social and Professional Aspects of Informatics

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Abílio Duarte Medeiros (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Francisco José Massa de Menezes (Regente) (T-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

•Posicionamento das tecnologias de informação nas empresas; evolução histórica dos conceitos e tecnologias envolvidas

•Ética profissional da Engenharia Informática e vantagens da adopção de um código ético

Saber Fazer:

•Comunicação eficaz, adaptada ao contexto

•Identificação de situações de conflito a nível profissional, gestão de conflitos

•Análise das controvérsias e impactos morais das tecnologias de informação, utilizando o “Disclosive method for Cyberethics”

Competências Complementares:

•Gestão do tempo e cumprimento de prazos

•Capacidades de apresentação oral e escrita

•Atitudes comportamentais que influenciam o sucesso de uma carreira Informática

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

•Integration of IT technologies in the enterprise; historical perspective of the relevant concepts and technologies

•Professional ethics; advantages of the adoption of an ethics code

Application:

- *Effective communication adapted to the different contexts*
 - *Identification of stress situations on professional level, conflict management*
 - *Analysis of controversies or moral impacts of information technology using the “Disclosive method for Cyberethics”*
- Soft-skills:**
- *Time management and meeting deadlines*
 - *Oral and written communication skills*
 - *Attitudes that affect career development*

3.3.5. Conteúdos programáticos:

- 1- *Perspectiva da evolução dos Sistemas de Informação na Sociedade*
- 2- *Perspectiva do papel dos Profissionais Informáticos na Sociedade*
- 3- *Atitudes comportamentais perante diferentes pessoas e organizações*
- 4- *Comunicação Oral, escrita e capacidade de síntese de um tema*
- 5- *Responsabilidades profissionais - Gestão do tempo e cumprimento de prazos*
- 6- *Identificação de situações limite no contexto profissional e como as abordar*
- 7- *Análise de evoluções de carreira*
- 8- *Introdução à Ética aplicada à Informática, segundo recomendações da ACM/IEEE-CS*
- 9- *Privacidade e Anonimato*
- 10- *Propriedade intelectual, direitos de autor e patentes*
- 11- *Pirataria informática e cibercrime*

3.3.5. Syllabus:

- 1 - *Evolution of Information Systems in Society*
- 2 - *Role of IT Professionals in Society*
- 3 - *Behavioral attitudes towards different people and organizations*
- 4 - *Oral and written communication, synthesis capacity of a theme*
- 5 - *Professional Responsibility - Time management and meeting deadlines*
- 6 - *Identification of extreme situations in a professional context and how to address them*
- 7 - *Career developments*
- 8 - *Introduction to Ethics applied to IT, according to recommendations of the ACM / IEEE-CS*
- 9 - *Privacy and Anonymity*
- 10 - *Intellectual property, copyright and patents*
- 11 - *Piracy and cybercrime*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A 1ª parte do programa (pontos 1-7) engloba aspectos relacionados com as funções e posicionamento de um profissional de informática numa empresa. É abordada a forma como as tecnologias de informação (TI) se integram numa empresa e como é que os paradigmas e tecnologias usados têm evoluído. Inclui-se também uma discussão sistemática de um conjunto de aptidões que um profissional de TI deve ter nomeadamente os relacionados com a comunicação oral e escrita, interação com colegas e gestão de carreira. Para consolidar estas aptidões, os estudantes fazem a análise e discussão de projectos de TI reais.

Uma 2ª parte (pontos 8-11) ocupa-se da ética profissional e dos impactos que as TI têm na sociedade; são discutidas várias questões controversas, como os relacionados com direitos de autor e da privacidade. Esta discussão é feita à luz de princípios éticos geralmente reconhecidos (IEEE/ACM). Esta actividade também auxilia a aquisição das competências complementares.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Part 1 (points 1 to 7 of syllabus) encompasses aspects related with the roles and positioning of an IT professional in a company. The integration of IT in companies is addressed and the how the paradigms and technologies used have evolved. This part also includes a sound presentation of the skills that an IT professional must have, in particular oral and writing communication, colleague and management relationships and career management. To consolidate these skills, students analyze and discuss several real cases related to cybertechnology.

Part 2 (points 8 to 11 of syllabus) takes up professional ethics and the impact that IT technologies have in society; Several controversial aspects are discussed, namely those related with copyright and rights of privacy. This discussion is made in the light of IEEE/ACM ethical principles; this also helps the acquisition of skills related to interaction with colleagues and superiors and communication appropriate to the context.

3.3.7. Metodologias de ensino (avaliação incluída):

Os temas teóricos são introduzidos de forma relativamente informal através de slides, textos, vídeos e/ou convidados. É em geral utilizado um caso prático que depois será objecto de análise em grupo ou individual, com discussões e apresentações orais. Esta reflexão começa na aula teórica, é continuada nas aulas práticas em grupos e dará origem a textos de síntese e apresentações feitas na aula.

Os elementos de avaliação a usar são

-dois testes (cada um com um peso de 25 %)

-dois trabalhos práticos em grupo, com um tema individual que inclui a produção de um pequeno relatório e uma apresentação. O trabalho realizado sobre a 1ª parte da matéria terá um peso (35%) mais elevado do que o 2º (15%). Nestes trabalhos os alunos usam as aptidões de comunicação oral e escrita e aplicam o "Disclosive Method for Cyberethics"

Os dois testes poderão ser substituídos pela realização de um exame de recurso.

3.3.7. Teaching methodologies (including evaluation):

The theoretical issues are introduced in a relatively informal means using slides, texts, videos and / or guests. It is generally used a case study which will then be examined in a group or individually, with discussions and oral presentations. This reflection on the lecture begins, is continued in group in practical classes and gives rise to text synthesis and presentations in class.

The elements of assessment are:

- Two tests (each weighing 25%)

- two practical work in groups, with an individual theme that includes the production of a short report and a presentation. The work done on the 1st part of the field will have a weight (35%) higher than the 2nd (15%). In these works the students use the skills of oral and written communication and apply the "Method for Disclosive Cyberethics"

The two tests may be replaced by a final exam.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os objectivos que têm a ver com o Saber e o Saber Fazer são avaliados através da realização dos testes escritos e também com a apreciação feita aos textos produzidos e apresentações realizadas, onde se avalia a correcta utilização do "Disclosive Method for Cyberethics".

A avaliação da forma como os objectivos relacionados com competências complementares são atingidos é feito usando um grelha em que se consideram todos os pontos referidos nos pontos 3, 4 e 5 do programa.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The Knowledge and Application objectives above are evaluated by conducting written tests and also by the assessment of the texts and presentations produced; this evaluation accesses the proper use of "Method for Disclosive Cyberethics" and the effectiveness of the oral and written communication skills.

The assessment of how the targets related to complementary competencies are achieved is done using an evaluation grid where all the aspects referred to in paragraphs 3, 4 and 5 of the syllabus are included.

3.3.9. Bibliografia principal:

IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices

<http://www.acm.org/about/se-code/>

Herman T. Tavani, Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing, Third Edition, 2011, Wiley

Michael J. Quinn, Ethics for the Information Age, Second Edition, 2006, Addison Wesley

Mapa IV - Empreendedorismo / Entrepreneurship

3.3.1. Unidade curricular:

Empreendedorismo / Entrepreneurship

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Virgílio António da Cruz Machado (TP-9h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Rogério Salema Araújo Puga Leal (TP-9h)

Fernanda Antonia Josefa Llussá (TP-9h)

Maria do Rosário de Meireles Ferreira Cabrita (TP-9h)

Ana Sofia Dinis Esteves (TP-9h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

O curso pretende motivar os alunos para o empreendedorismo e para a necessidade da inovação tecnológica. O programa cobre vários tópicos que são importantes para a adoção de uma cultura aberta aos riscos suscitados em processos de criação de novos produtos ou atividades que exigem características empreendedoras.

No final desta unidade curricular, os alunos deverão ter desenvolvido um espírito empreendedor, uma atitude de trabalho em equipa e estar aptos a:

- 1) Identificar ideias e oportunidades para empreenderem novos projetos;*
- 2) Conhecer os aspetos técnicos e organizacionais inerentes ao lançamento dos projetos empreendedores;*
- 3) Compreender os desafios de implementação dos projetos (ex: mercado, financiamento, gestão da equipa) e encontrar os meios para os ultrapassar;*
- 4) Expor a sua ideia e convencer os stakeholders.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This course is intended to motivate students for entrepreneurship and the need for technological innovation. It covers a list of topics and tools that are important for new venture creation as well as for the development of creative initiatives within existing enterprises. Students are expected to develop an entrepreneurship culture, including the following skills:

- 1) To identify ideas and opportunities to launch new projects;*
- 2) To get knowledge on how to deal with technical and organizational issues required to launch entrepreneurial projects;*
- 3) To understand the project implementation challenges, namely venture capital and teamwork management, and find the right tools to implement it;*
- 4) To show and explain ideas and to convince stakeholders.*

3.3.5. Conteúdos programáticos:

O empreendedorismo como estratégia de desenvolvimento pessoal e organizacional. Processos de criação de ideias. A proteção da propriedade intelectual: patentes e formalismos técnicos. A gestão de um projeto de empreendedorismo: planeamento; comunicação e motivação; liderança e gestão de equipas Marketing e inovação para o desenvolvimento de novos produtos e negócios. O plano de negócios e o estudo técnicofinanceiro.

Financiamento e Sistemas de Incentivos: formalidades e formalismos. A gestão do crescimento e o intra-empreendedorismo.

3.3.5. Syllabus:

Strategy for entrepreneurship. Ideation and processes for the creation of new ideas. Industrial property rights and protection: patents and technical formalities. Managing an entrepreneurial project: planning; communication and motivation; leadership and team work. Marketing and innovation for the development of new products and businesses. Business plan and entrepreneurial finance. System of Incentives for young entrepreneurs. Managing growth and intrapreneurship.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O conteúdo programático foi desenhado para incentivar o aluno ao empreendedorismo e à percepção e análise da envolvente em busca de oportunidades de negócio, de forma a que consiga aplicar os conhecimentos adquiridos:

- 1) na transformação de conhecimento científico em ideias de negócio;*
- 2) na criação, seleção e desenvolvimento de uma ideia para um novo produto ou serviço;*
- 3) na elaboração de um plano de negócio e de um plano de marketing;*
- 4) na exposição das suas ideias em curto tempo e em ambientes stressantes.*

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus was designed to encourage the student for entrepreneurship and for the perception and analysis of new business opportunities; with this program, the student may apply the knowledge provided:

- 1) to transform scientific knowledge in business ideas;*
- 2) to create, select and develop an idea for a new product or service;*
- 3) to draw a business plan and a marketing plan*
- 4) to better explain and present its ideas in a short time and stressed environments.*

3.3.7. Metodologias de ensino (avaliação incluída):

Este curso será ministrado a alunos dos 4º ou 5º anos dos programas de Mestrado integrado e de 2º ciclo. O programa é dimensionado para decorrer entre o 1º e o 2º semestre, num período de 5 semanas, envolvendo um total de 45 horas presenciais (TP), organizadas em 15 sessões de 3 horas e exigindo um esforço global de 3 ECTS.

As aulas presenciais baseiam-se na exposição dos conteúdos do programa. Os estudantes serão solicitados a aplicar as competências adquiridas através da criação e desenvolvimento de uma ideia (produto ou negócio). As aulas integrarão alunos provenientes de diversos cursos com vista a promover a integração de conhecimento derivado de várias áreas científicas e envolverão professores e "mentores" com background diverso em engenharia, ciência, gestão e negócios.

A avaliação compreende a apresentação e defesa da ideia num elevator pitch e do respetivo relatório (realizado em grupo de 4-5 elementos). A apresentação contribuirá com 60% e o relatório com 40% para a nota final.

3.3.7. Teaching methodologies (including evaluation):

This course is directed to students from the 4th and 5th years of the "Mestrado Integrado" (Integrated Master) and students from the 2nd cycle (Master).

The program was designed for a duration of 5 weeks, with a total of 45 hours in class (15 sessions of 3 hours each) - 3 ECTS.

Classes are based in an exposition methodology. Students will be asked to apply their skills in the creation and development of an idea, regarding a new product or a new business.

Classes integrate students from different study programs to promote the integration of knowledge derived from various scientific areas and involve academic staff and "mentors" with diverse background in engineering, science, management and business.

Students evaluation is based on the development and presentation of an idea/project in an elevator pitch, and its report. The work should be developed in teams of 4-5 members. The presentation should account for 60% of the final mark and the report 40%.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Considerando o tempo disponível (5 semanas), a metodologia de ensino preconiza que em cada semana sejam discutidos e trabalhados (em grupo) os temas apresentados, os quais tinham sido definidos nos objetivos de aprendizagem. Na 1ª semana os temas a abordar estão relacionados com os aspetos estratégicos do empreendedorismo, a geração de ideias, a liderança e a gestão de equipas; como resultado os alunos deverão constituir e organizar as suas equipas para poderem definir o problema que se pretende resolver. Na 2ª semana, os temas apresentados permitirão que o aluno possa evoluir no seu projeto acrescentando opções de soluções ao problema identificado na semana anterior e proceder à seleção de uma delas. Na 3ª semana, a abordagem ao mercado e às condições de comercialização viabilizarão a concretização do plano de marketing. Na 4ª semana, abordar-se-ão os aspetos relacionados com a viabilidade financeira do projeto, possibilitando a realização do respetivo plano de negócio e do seu financiamento. Na última semana, abordar-se-á o processo de exposição da ideia aos potenciais interessados, tendo os alunos que realizar a apresentação e defesa do seu projeto num elevator pitch, perante um júri.

Neste sentido, a metodologia privilegia

- 1) a apresentação de casos práticos e de sucesso;*
- 2) a promoção de competências nos domínios comportamentais, nomeadamente, no que respeita ao desenvolvimento do sentido crítico, à defesa de ideias e argumentos baseados em dados técnico-científicos, à tolerância e capacidade de gestão de conflitos em situações adversas e stressantes.*
- 3) a participação dos alunos nos trabalhos colocados ao longo da unidade curricular e a sua apresentação.*

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Considering the available time (5 weeks), the teaching methodology praisers that, in each week, the subjects presented and defined in the learning objectives are discussed and worked (in groups).

In the first week, the subjects introduced to students are related with entrepreneurial strategic issues, generation of ideas, leadership and work team management; as a result, the students will have to organize their teams to be able to define the problem. In the 2nd week, the subjects presented will allow the student to pursuit

its project; they have to consider different options for the problem identified in the previous week. In the 3rd week, the market related issues are approached, and the students are asked to build a marketing plan. In the 4th week, financial issues are addressed, making it possible to accomplish a business plan. In the last week, the process of how to expose the idea to potential stakeholders is addressed; the students are required to present and argue their project in an elevator pitch.

This methodology gives priority to:

- 1) the presentation of practical and successful cases;*
- 2) the promotion of soft skills, namely, in what concerns to the development of critical thinking, the defense of ideas and arguments based on technical-scientific data, to the tolerance and capacity of dealing with conflicts in adverse and stressful situations.*
- 3) the participation of the students in practical works and assessments and their presentation.*

3.3.9. Bibliografia principal:

Books

Burns, P. (2010). Entrepreneurship and Small Business: Start-up, Growth and Maturity, Palgrave Macmillan, 3rd Ed.

Kotler, P. (2011). Marketing Management, Prentice-Hall

Shriberg, A. & Shriberg (2010). Practicing Leadership: Principles and Applications, John Wiley & Sons, 4th Ed.

Spinelli, S. & Rob Adams (2012). New Venture Creation: Entrepreneurship for the 21st Century. McGraw-Hill, 9th Ed.

Byers, Thomas H., Dorf R. C., Nelson, A. (2010). Technology Ventures: From Idea to Enterprise, 3rd Ed., McGraw-Hill

Hisrich, R. D. (2009). International Entrepreneurship: Starting, Developing, and Managing a Global Venture, Sage Publications, Inc

Hisrich, R.D., Peters, M. P., Shepherd, D.A. Entrepreneurship, 7th Ed., McGraw-Hill, 2007

Journals

Entrepreneurship Theory and Practice

Journal of Entrepreneurship

International Entrepreneurship and Management Journal

International Journal of Entrepreneurial Behaviour & Research

Entrepreneurship & Regional Development

Journal of Business Venturing

Mapa IV - Dissertação / Dissertation

3.3.1. Unidade curricular:

Dissertação / Dissertation

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Manuel Correa Calvente Barahona (OT-40h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Todos os docentes da área científica principal do curso de Mestrado Integrado em Engenharia Informática (OT-40h) / All professors in the main scientific area of the Integrated Master in Computer Science and Engineering (40h).

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

O trabalho conducente à dissertação de Mestrado Integrado em Engenharia Informática enquadra-se nos termos definidos na alínea b) do Artigo 20º do Decreto Lei 74/2006, e consiste num trabalho individual de investigação e/ou desenvolvimento que explora os conhecimentos adquiridos ao longo do curso, original e especialmente concebido para esse fim. O trabalho de dissertação pode basear-se na elaboração de uma tese de natureza científica ou no desenvolvimento de um projeto avançado de engenharia de concepção, ou combinando as duas vertentes, avaliado de acordo com parâmetros uniformes.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The dissertation work for the integrated Master in Computer Science and Engineering conforms with the requirements of item b) Art 20º of DL 74/2006, and consists of an individual research and / or development work which explores the knowledge acquired in the whole program, original, and specifically identified for that

purpose. The dissertation work may be based on the elaboration of a research-oriented thesis of scientific nature, or in the development of an advanced conception-oriented engineering project, or combining the two orientations, and evaluated according to common criteria.

3.3.5. Conteúdos programáticos:

Desenvolvimento de trabalho conducente a elaboração de dissertação de mestrado em Engenharia Informática.

3.3.5. Syllabus:

Development of the work leading to the elaboration of a master dissertation in Computer Science and Engineering.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos são naturalmente coerentes com os objetivos, dada a sua natureza especial da unidade curricular.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Given the special nature of the curricular unit, the syllabus are naturally coherent with the unit's objectives

3.3.7. Metodologias de ensino (avaliação incluída):

O trabalho conducente à dissertação de mestrado desenrola-se em duas fases. Na primeira fase (com a duração de 1 semestre, mas a tempo parcial de aproximadamente 40%) o estudante desenvolve o trabalho introdutório do tema da sua dissertação e apresenta um relatório incluindo um plano do trabalho a realizar para concluir a Dissertação do Mestrado, durante a segunda fase. Esse relatório é discutido e avaliado numa sessão pública por um júri intermédio que o deverá validar e eventualmente apresentar sugestões relevantes. Segue-se a segunda fase (com a duração de 1 semestre, a tempo inteiro). O trabalho desenvolvido será formalmente apresentado na Dissertação de Mestrado, defendida em provas públicas perante o júri, que a avalia e lhe atribui uma nota final, de acordo com o regulamento geral da FCT e da UNL.

3.3.7. Teaching methodologies (including evaluation):

The work leading to the master dissertation develops in two consecutive phases. In the first phase (with the duration of one term, but in partial time of approximately 40%) the student develops preliminary work for the dissertation theme and presents a written report, including a plan of the work needed to conclude the master dissertation work. This report is discussed and evaluated in a public session by an interim jury that validates and possibly proposes suggestions for improving the future work. Follows the second phase (duration of one term, with full time dedication). The work developed is formally reported in the Master Dissertation, which is defended by the candidate in public exam before a jury who assigns the final grading, according to the general regulations of FCT and UNL.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Tendo em conta a natureza especial desta unidade curricular, decorre que existe coerência adequada entre as metodologias de ensino e os objetivos de aprendizagem da mesma. Acresce que a metodologia de ensino adotada, estruturada em duas fases, com um ponto de aferição intermédio após a fase de preparação, e que resulta da experiência de vários anos do Departamento de Informática na orientação de dissertações de mestrado, tem-se revelado um fator promotor do aumento de qualidade do processo de desenvolvimento das dissertações, na opinião de todos os envolvidos (estudantes, docentes, membros de júri, e parceiros exteriores).

As duas fases combinam-se de forma integrada na unidade curricular Dissertação, de 42 ECTS, sendo que a segunda fase do trabalho, correspondente à elaboração da dissertação e estimada em 30 ECTS será tipicamente desenvolvida pelo estudante em dedicação total. O trabalho de dissertação, independentemente da sua natureza de introdução à investigação ou de projeto avançado de engenharia de conceção, poderá desenvolver-se quer em contexto académico, quer em contexto de colaboração académico-empresarial, em qualquer dos casos, sob coordenação científica geral da comissão científica de curso e orientação formal de um membro do corpo docente do curso. Tipicamente, serão avaliados os seguintes parâmetros:

- 1. Qualidade da análise do estado da arte e da sua adequação aos objetivos do trabalho;*
- 2. Qualidade do trabalho realizado;*
- 3. Extensão do trabalho realizado;*
- 4. Qualidade da análise crítica da contribuição e dos resultados obtidos;*
- 5. Qualidade da apresentação oral e da argumentação;*

6. *Qualidade do Relatório;*

7. *Apreciação Geral.*

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Having in mind the special nature of this curricular unit, it clearly results that there is adequate coherence between the teaching methodologies and the learning outcomes. Additionally, we notice that the teaching methodology, structured in two phases, with an interim assessment point after the preparation first phase, which results from the several years experience of the Department on supervising master dissertations, has revealed itself as a factor that promotes a quality increase of the dissertation supervision process, in the opinion of all the stakeholders (students, faculty members, jury members, and external collaborators). The two phases combine in an integrate way in the curricular unit Dissertation, of 42 ECTS units, where the second phase, corresponding to the elaboration of the planned dissertation work and estimated in 30 ECTS is typically developed in full time dedication by the student. The dissertation work, independently of its orientation towards research or advanced engineering project, may be developed either in academic context or in some collaboration between the department and an external institution (such as a firm), in any case under the scientific coordination of the program's scientific committee and formal supervision of a faculty member. Typically, the following parameters are taken into account:

1. *Quality of the state-of-the-art review and its suitability to the work objectives;*
2. *Quality of the performed work;*
3. *Extension of the performed work;*
4. *Quality of the critical analysis of the contributions and results obtained;*
5. *Quality of the oral presentation and argumentation;*
6. *Quality of the written document;*
7. *Global appreciation.*

3.3.9. Bibliografia principal:

Depende de cada trabalho de dissertação / Depends on each specific dissertation work.

Mapa IV - Análise e Desenho de Algoritmos / Design and Analysis of Algorithms

3.3.1. Unidade curricular:

Análise e Desenho de Algoritmos / Design and Analysis of Algorithms

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Margarida Paula Neves Mamede (Regente) (T-28h; P-84h)

Fernanda Maria Barquinha Tavares Vieira Barbosa (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber: Conhecer os algoritmos fundamentais de grafos, os tipos abstratos de dados envolvidos e as estruturas de dados usadas para os implementar com eficiência. Compreender complexidade amortizada. Definir e identificar três técnicas de desenho de algoritmos: estratégias greedy, programação dinâmica, e transformação e conquista.

Saber Fazer: Formalizar um problema concreto em termos de grafos e adaptar um algoritmo clássico para o resolver. Escolher, comparar, adaptar e utilizar estruturas de dados adequadas ao problema a resolver.

Calcular a complexidade de algoritmos com base na complexidade amortizada das funções auxiliares e calcular a complexidade amortizada dessas funções. Avaliar soluções e efetuar escolhas fundamentadas.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge: Know the fundamental graph algorithms, the required abstract data types and the data structures used to implement them efficiently. Understand amortized analysis. Define and identify three algorithm design techniques: greedy strategies, dynamic programming and transform-and-conquer.

Application: Formulate a clean graph problem from a real-world problem and adapt a classical algorithm to solve it. Choose, compare, adapt, and use suitable data structures for a given problem. Calculate the running

time of an algorithm based on the amortized running times of the inner functions and perform their amortized analysis. Evaluate solutions and justify choices.

3.3.5. Conteúdos programáticos:

1. *Introdução ao estudo de grafos. Definições fundamentais. Tipos abstratos de dados grafo não orientado e grafo orientado. Implementações de grafos.*
2. *Algoritmos elementares de grafos. Percursos em profundidade e em largura. Ordenação topológica. Teste à aciclicidade.*
3. *Árvores mínimas de cobertura. Algoritmo de Kruskal. Tipo abstrato de dados partição.*
4. *Complexidade amortizada. Métodos da contabilidade e do potencial.*
5. *Algoritmo de Prim. Tipo abstrato de dados fila com prioridade adaptável. Filas binomiais e de Fibonacci.*
6. *Caminhos mais curtos. Algoritmos de Dijkstra, Bellman-Ford e Floyd-Warshall.*
7. *Fluxos máximos. Método de Ford-Fulkerson. Algoritmo de Edmonds-Karp. Emparelhamentos máximos em grafos bipartidos. Cortes mínimos.*

3.3.5. Syllabus:

1. *Introduction to the study of graphs. Fundamental definitions. The abstract data types undirected graph and directed graph. Implementations of graphs.*
2. *Elementary graph algorithms. Depth-first and breadth-first traversals. Topological sort. Test for acyclicity.*
3. *Minimum spanning trees. Kruskal's algorithm. The disjoint sets abstract data type.*
4. *Amortized analysis. The accounting and the potential methods.*
5. *Prim's algorithm. The adaptable priority queue abstract data type. Binomial and Fibonacci queues.*
6. *Shortest paths. The algorithms of Dijkstra, Bellman-Ford, and Floyd-Warshall.*
7. *Maximum flows. The Ford-Fulkerson method. The Edmonds-Karp algorithm. Maximum bipartite matchings. Minimum cuts.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular podem ser sintetizados da seguinte forma.

1. *Conhecer os algoritmos fundamentais de grafos e as suas implementações eficientes: Os pontos 2, 3, 5, 6 e 7 dos conteúdos programáticos cobrem os algoritmos, os novos tipos abstratos de dados requeridos e as novas estruturas de dados que permitem implementações eficientes. (Novo significa que não é estudado na primeira unidade curricular de Algoritmos e Estruturas de Dados.)*
2. *Calcular complexidades amortizadas, avaliar soluções e efetuar escolhas fundamentadas: A complexidade amortizada, que é introduzida e exemplificada no ponto 4, é necessária para se poderem analisar (avaliar) todos os algoritmos apresentados.*
3. *Definir e identificar as técnicas de desenho de algoritmos greedy, programação dinâmica, e transformação e conquista: Estas técnicas são explicadas nos pontos 3, 5, 6 e 7, quando são usadas para desenhar alguns dos algoritmos estudados.*

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The unit objectives can be summarized in the following way.

1. *To know the fundamental graph algorithms and their efficient implementations: Syllabus topics 2, 3, 5, 6, and 7 cover the algorithms, the new required abstract data types and the new data structures that allow efficient implementations. (New means that it is not studied in the first Algorithms and Data Structures unit).*
2. *Perform amortized analyses, evaluate solutions and justify choices: Amortized analysis, which is introduced and exemplified in topic 4, is needed to be able to analyse (evaluate) all presented algorithms.*
3. *Define and identify the algorithm design techniques greedy, dynamic programming and transform-and-conquer: These techniques are explained in topics 3, 5, 6 and 7, when they are used to design some of the studied algorithms.*

3.3.7. Metodologias de ensino (avaliação incluída):

O ensino consiste na exposição da matéria em aulas teóricas e na resolução de problemas em aulas práticas de laboratório. No laboratório, os alunos desenham, analisam e implementam algoritmos.

A avaliação é composta por quatro trabalhos de grupo e dois testes. Cada trabalho consiste no desenho, análise e implementação de um algoritmo para resolver um problema de um concurso de programação (e.g., SWERC), na elaboração de um relatório e na realização de uma discussão. Os alunos têm de fazer três (dos quatro) trabalhos. Se a média dos testes não for positiva, os alunos podem realizar um exame final. Os testes e o exame são com consulta.

Condição para obter aprovação: $NotaP \geq 9.5$ e $NotaT \geq 9.5$, onde $NotaP$ é a média das notas dos trabalhos e $NotaT$ é a média das notas dos testes ou a nota do exame.

3.3.7. Teaching methodologies (including evaluation):

There are two lectures and a lab session each week. In the laboratory, students design, analyse and implement algorithms.

Assessment comprises four group programming projects and two tests. Each programming project consists in the design, analysis and implementation of an algorithm for solving a programming contest (e.g., SWERC) problem, in the writing of a report and in a discussion. Students must do three (of the four) programming projects. If the mean of the test grades is less than 9.5 (out of 20), students can do a final exam. The tests and the exam are open-book.

Condition to succeed: $Pgrade \geq 9.5$ and $Tgrade \geq 9.5$, where $Pgrade$ is the mean of the project grades and $Tgrade$ is the mean of the test grades or the exam grade.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Em cada aula prática, à exceção da última, e em cada trabalho de grupo, os alunos resolvem um problema de um concurso de programação (SWERC, MIUP, etc.). Para resolver todos os problemas, é necessário adaptar, pelo menos uma vez, cada um dos algoritmos dados nas aulas teóricas. Sempre que possível, a ordem dos problemas não respeita a ordem pela qual a matéria é lecionada.

Utilizar problemas de concursos de programação tem quatro grandes vantagens. Em primeiro lugar, aumenta substancialmente a motivação dos alunos para os resolver. Em segundo lugar, dificulta a identificação da forma de os atacar, porque eles estão sempre especificados como problemas do mundo real. Em terceiro lugar, como é consensualmente aceite que um problema de um concurso de programação só está resolvido quando o programa é aceite por um sistema automático de teste (Mooshak), é necessário desenhar e analisar adequadamente o algoritmo antes de o implementar, para que os tempos de execução e a memória requerida não excedam os limites permitidos. Por último, mas não menos importante, os testes utilizados para aceitar ou rejeitar programas nos concursos de programação são muito completos mas nunca são divulgados. Consequentemente, os alunos são obrigados a fabricar os seus próprios testes, desenvolvendo a aptidão para executar uma das tarefas cruciais da produção de código: o teste de programas.

No caso dos trabalhos de grupo, os alunos elaboram um relatório sobre o desenho, a análise e a implementação do algoritmo desenvolvido, especificando e justificando todas as opções tomadas. Os prazos para submeter o programa ao sistema automático de teste e para entregar o relatório têm de ser rigorosamente cumpridos. Portanto, os alunos desenvolvem a aptidão para avaliar soluções e efetuar escolhas fundamentadas e capacidades de comunicação escrita e gestão do tempo.

Como em qualquer processo de aprendizagem, é fundamental que os alunos recebam feedback sobre o trabalho que vão realizando. Por esse motivo, os docentes identificam os pontos fortes e fracos de cada programa, no final das aulas práticas ou logo após os trabalhos terem sido avaliados.

Na última aula prática resolvem-se exercícios de papel e lápis para consolidar o cálculo da complexidade amortizada de algoritmos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In each lab session except for the last one and in each group programming project, students solve a programming contest (SWERC, MIUP, etc.) problem. To solve all problems, every graph algorithm presented in the lectures must be adapted at least once. Whenever possible, problems do not match the order in which material is introduced.

Making use of programming contest problems has four advantages. First of all, students are much more motivated to solve them. In the second place, it is more difficult to identify the way to tackle them, because they are always stated as real-world problems. In the third place, as it is consensually considered that a programming contest problem is solved only when the program is accepted by an automatic judging system (Mooshak), the algorithm has to be suitably designed and analysed before being implemented so that running times and memory requirements do not exceed the allowed limits. Last but not least, tests used to accept or reject programs in programming contests are very complete but they are never made available. Consequently, students must make their own tests, improving skills in software testing.

For every group programming project, students write a report on the design, analysis and implementation of the developed algorithm, specifying and justifying all choices made. The deadlines for submitting programs to the automatic judging system and for delivering reports are strict. So, students develop the ability to evaluate solutions and justify choices, and skills in writing communication and time management.

As in any learning process, students have to receive feedback on their work. For this reason, teachers identify the strong and the weak points of each program, at the end of lab sessions or just after the evaluation of the programming projects.

In the last lab session, students solve pencil and paper exercises that aim at consolidating the use of amortized analysis.

3.3.9. Bibliografia principal:

Main References:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3rd edition)*. The MIT Press, 2009.
2. Jon Kleinberg and Éva Tardos. *Algorithm Design*. Addison-Wesley, 2006.

Complementary References:

3. Anany Levitin. *Introduction to The Design and Analysis of Algorithms (3rd edition)*. Addison-Wesley, 2011.
4. Steven S. Skiena. *The Algorithm Design Manual (2nd edition)*. Springer, 2008.
5. Steven S. Skiena and Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer, 2003.

Mapa IV - Aprendizagem Automática / Machine Learning

3.3.1. Unidade curricular:

Aprendizagem Automática / Machine Learning

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Manuel Corrêa Calvente Barahona (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Susana Maria dos Santos Nascimento Martins de Almeida (Regente) (T-14h; P-28h)
Ludwig Krippahl (T-14h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Compreender os paradigmas e desafios da área de Aprendizagem Automática. Aprendizagem Supervisionada, Aprendizagem Não-Supervisionada e Aprendizagem por Reforço.*
- *Aprender métodos fundamentais e suas aplicações na descoberta de conhecimento orientada aos dados. Dados, seleção de modelos, complexidade de modelos, etc.*
- *Compreender vantagens e limitações dos métodos de Aprendizagem Automática estudados.*

Saber Fazer:

- *Implementar e adaptar algoritmos de Aprendizagem Automática.*
- *Modelar experimentalmente dados reais usando esses algoritmos.*
- *Interpretar e avaliar resultados experimentais.*
- *Validar e comparar algoritmos de Aprendizagem Automática.*

Competências Complementares:

- *Capacidade de avaliar a adequação dos métodos a dados e aplicações práticas.*
- *Capacidade de avaliar criticamente os resultados obtidos.*
- *Autonomia para aplicar e aprofundar os conhecimentos na área de Aprendizagem Automática.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Understand the paradigms and challenges of Machine Learning, distinguishing Supervised, Unsupervised and Reinforcement learning.*
- *Learn the fundamental methods and their applications in data oriented knowledge discovery. Understand data*

features, the selection of models and their complexity.

- *Understand the advantages and disadvantages of the different methods.*

Application:

- *Implement and adapt Machine Learning algorithms;*
- *Model real data experimentally.*
- *Interpret and evaluate experimental results.*
- *Validate and compare different Machine Learning algorithms.*

Soft-skills:

- *Evaluate the suitability of each method to concrete applications and data sets.*
- *Critical evaluation of the results.*
- *Autonomy and self-reliance in the application and furthering studies in Machine Learning.*

3.3.5. Conteúdos programáticos:

1. Introdução à Aprendizagem Automática.

1.1 Aprendizagem Supervisionada, Aprendizagem Não-Supervisionada e Aprendizagem por Reforço.

1.2 Classificação, regressão e agrupamento.

2. Dados

2.1 Tipos de dados.

2.2 Medidas de proximidade e medidas de dispersão de dados.

2.3 Tópicos de pré-processamento e visualização de dados

2.4 Redução de dimensionalidade por Análise de Componentes Principais

3. Aprendizagem Supervisionada

3.1 Regressão

3.2 Árvores de Decisão

3.3 Redes Neurais

3.4 Máquinas de Suporte Vetorial

3.5 Modelos gráficos

3.6 Classificador dos K-Vizinhos mais Próximos

3.7 Avaliação e comparação de métodos de classificação

3.8 Ensembles

4. Aprendizagem Não-Supervisionada

4.1 Métodos de Agrupamento por Partição

4.2 Métodos de Agrupamento Hierárquico

4.3 Mapas Auto-Organizados

4.4 Métodos de Agrupamento Probabilístico

4.5 Métodos de Agrupamento Difuso

4.6 Avaliação de métodos e de resultados de agrupamento

3.3.5. Syllabus:

1. Introduction to Machine Learning

1.1 Supervised Learning, Unsupervised Learning and Reinforcement Learning.

1.2 Classification, regression and clustering.

2. Data

2.1 Types of Data

2.2 Measures of similarity and dissimilarity

2.3 Introduction to data preprocessing and visualization

2.4 Dimensionality reduction by Principal Component Analysis

3. Supervised Learning

3.1 Regression

3.2 Decision Trees

3.3 Artificial Neural Networks

3.4 Support Vector Machines

3.5 Graphical models

3.6 K-nearest neighbour classifier

3.7 Methods for classifier evaluation and comparison

3.8 Ensembles

4. Unsupervised Learning

4.1 Partitional clustering

4.2 Hierarchical clustering

4.3 Self organizing maps

4.4 Probabilistic clustering

4.5 Fuzzy clustering

4.6. Clustering evaluation

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O programa da disciplina, tanto na componente de Aprendizagem Supervisionada como na componente de Aprendizagem Não-Supervisionada, cobre todos os tópicos base e uma parte substancial dos tópicos considerados opcionais na subárea 'Advanced Machine Learning' [elective] do Computer Science Curriculum 2013 da ACM (<http://cs2013.org/>).

As várias metodologias estudadas são enquadradas com exemplos de aplicação.

Os trabalhos práticos e exercícios tutoriais permitem consolidar conhecimentos na implementação/adaptação dos algoritmos estudados a problemas reais. Será dada particular atenção à interpretação de resultados experimentais, avaliação dos métodos usados e comparação de modelos.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Both the Supervised and Unsupervised Learning components of the syllabus cover all the core subjects and most of the elective subjects for Machine Learning in the ACM Computer Science Curricula 2013 (<http://cs2013.org/>).

The methods studied are applied to practical examples. Tutorial classes and practical assignments promote the consolidation of theoretical knowledge as well as practice in the implementation and adaptation of the algorithms to real life problems. Practical assignments also focus on the interpretation of experimental results and the evaluation of the methods used, including the comparison of different models.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas são lecionados os tópicos fundamentais da matéria, os quais deverão ser complementados com a leitura da bibliografia aconselhada. Nas aulas teóricas será dedicado tempo para exposição da matéria e para dúvidas e discussão com os alunos.

As aulas laboratoriais destinam-se à orientação tutorial e realização dos trabalhos práticos sobre tópicos selecionados da matéria.

Serão disponibilizados os acetatos da matéria teórica.

Será disponibilizada uma página Web da disciplina onde se mantém informação atualizada sobre o funcionamento da mesma.

Avaliação

A avaliação da disciplina é composta por 2 testes escritos individuais e 2 trabalhos práticos de grupo.

3.3.7. Teaching methodologies (including evaluation):

Lectures will cover the fundamental topics of the subject matter, which the students should complement with the given bibliography. All lecture materials will be supplied for further study. Lectures will include some time for questions and discussion of the subject matter.

Tutorial classes will be dedicated to exercises and guidance in the practical assignments, focusing on selected topics.

Class schedules and materials will be supplied online, as well as additional information regarding the course.

Assessment

Students will be evaluated on 2 written individual tests and 2 group assignments.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os objetivos desta unidade curricular são:

a) a aquisição de conhecimentos teóricos para a compreensão dos vários paradigmas de Aprendizagem

- Automática, das metodologias e dos princípios da sua aplicação;*
b) a aptidão para implementar estes algoritmos e, assim, resolver problemas práticos;
c) a competência para avaliar a adequação de cada método a cada caso particular e para avaliar criticamente os resultados obtidos.

O objetivo a) será atingido pela exposição da matéria, discussão e esclarecimento de dúvidas nas aulas teóricas, onde se abordará o fundamento teórico desta unidade curricular. Esta matéria será avaliada em testes individuais, onde os alunos têm um tempo limitado para mostrar o seu grau de assimilação e compreensão dos assuntos.

Os objetivos b) e c) serão atingidos principalmente pela componente prática da disciplina, se bem que sempre assente nas aulas teóricas, por meio das aulas práticas e do trabalho autónomo do aluno. As aulas práticas darão aos alunos oportunidade para aplicar os conhecimentos a casos concretos, experimentar as diferentes abordagens e examinar, em detalhe, os vários algoritmos em condições realistas. A componente prática será avaliada em dois trabalhos de grupo, avaliação esta que focará não só a implementação e aplicação prática dos algoritmos mas também a capacidade crítica de avaliação dos métodos escolhidos e dos resultados. Esta componente prática contribuirá também para a autonomia dos alunos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The course objectives are:

- a) the acquisition of theoretical knowledge to allow the understanding of the different Machine Learning paradigms, methodologies and their principles of application;*
b) the skill to implement these algorithms and, thus, solve practical problems;
c) the competence to evaluate the suitability of each method to each specific situation and for a critical evaluation of the results.

Objective a) will be reached mainly through the theoretical lectures, which include exposition of the subject matter and discussion. These lectures will cover the theoretical foundation of the course and this knowledge will be evaluated in individual written tests, where the students will have limited time to show their success in assimilating and understanding the subjects.

Objectives b) and c) will be reached mainly through practice and the practical assignments, during the tutorial classes and the student's own work. Tutorial classes will give the students the opportunity to apply their knowledge to specific cases, experiment with different approaches and examine, in detail, the different algorithms in realistic conditions. This practical component will be evaluated in two group assignments, focusing on the implementation and practical application of the algorithms but also on the critical assessment of the methods and results. This component will contribute to give the student the autonomy and self-reliance needed for future applications and further studies in this field.

3.3.9. Bibliografia principal:

- Bishop, C. M.. *Pattern Recognition and Machine Learning*, Springer, 2006.
- Duda, R., Hart, P. and Stork, D., *Pattern Classification, Second Edition*, John Wiley & Sons, Inc., 2001.
- Mitchell, T.M. *Machine Learning*, McGraw-Hill, 1997.

Mapa IV - Concorrência e Paralelismo / Concurrency and Paralelism

3.3.1. Unidade curricular:

Concorrência e Paralelismo / Concurrency and Paralelism

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Alberto Cardoso e Cunha (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Pedro Abílio Duarte de Medeiros (Regente) (T-14h)

Rodrigo Seromenho Miragaia Rodrigues (T-14h),

Maria Cecília Farias Lorga Gomes (P-56h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Esta é uma unidade curricular do bloco de consolidação do curso. O seu principal objetivo é dar uma panorâmica das abstrações e ferramentas usadas na programação de sistemas com múltiplos processadores. Saber:

- *Identificar os modelos para a resolução de problemas em arquiteturas com múltiplos processadores*

- *Conhecer os paradigmas usados no desenvolvimento de algoritmos para as máquinas referidas*
- *Avaliar a correção e o desempenho de algoritmos concorrentes e paralelos*
- *Conhecer linguagens, bibliotecas e ferramentas usadas no desenvolvimento de programas concorrentes e paralelos*

Saber Fazer:

- *Subdividir um problema de forma a adaptá-lo à resolução de forma eficiente*
- *Utilizar a linguagem Java no desenvolvimento de aplicações concorrentes*
- *Contactar com as ferramentas usadas no ciclo de desenvolvimento de uma aplicação concorrente, nas fases de conceção, implementação, depuração de comportamento e de desempenho e no "deployment"*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The course is included in the consolidation block of the degree. Its main objective is to give an overview of the abstractions and tools used in multiprocessor programming.

Knowledge:

- *To identify the models used for problem solving in multiprocessors*
- *To know the paradigms used in the development of algorithms for multiprocessors*
- *To evaluate the correction and performance of concurrent and parallel algorithms*
- *To know the languages, libraries and tools used in the development of concurrent and parallel programs*

Application:

- *Partitioning a problem to execute it efficiently on a multiprocessor*
- *To use the Java language for developing concurrent applications*
- *To make contact with the tools used in the development of a concurrent and parallel application, during the design, implementation, debugging and deployment stages*

3.3.5. Conteúdos programáticos:

1. Introdução à execução concorrente

1.1 Abstrações básicas e propriedades da execução concorrente.

1.2 Corridas e interblocagem. Técnicas para as evitar e detetar.

2. Multiprocessadores.

2.1 Taxonomia das arquiteturas de multiprocessadores.

2.2 Coerência de caches em multiprocessadores de memória partilhada.

2.3 Aceleração. Lei de Amdahl.

3. Programação concorrente em memória partilhada.

3.1 Mecanismos básicos de sincronização em memória partilhada.

3.2 Monitores.

3.3 Estruturas de dados concorrentes. Caso de estudo: listas.

4. Transações e Memória Transacional.

4.1 Conceito de transação.

4.2 Controlo de concorrência: princípios e algoritmos.

4.3 Memória transacional em software. Suporte de hardware.

5. Programação paralela.

5.1 Mecanismos básicos de sincronização e comunicação em memória distribuída.

5.2 Estratégias de decomposição.

5.3 Padrões algorítmicos.

5.4 Modelos de programação e seu mapeamento.

5.5 Caso de estudo do modelo Map-Reduce.

3.3.5. Syllabus:

1. Introduction to concurrent execution.

1.1. Properties and basic abstractions of concurrent execution.

1.2. Races and deadlocks. Techniques to avoid them.

2. Multiprocessors.

2.1. Taxonomy of multiprocessor architectures.

2.2. Cache coherence in shared memory multiprocessors.

2.3. Speedup. Amdahl's law.

3. Concurrent programming with shared memory.

3.1. Basic mechanisms for shared memory synchronization.

3.2. Monitors.

3.3. Concurrent data structures. Case study: lists.

4. Transactions and transactional memory.

4.1. Concept of transaction.

4.2. Concurrency control: principles and algorithms.

4.3. Software transactional memory. Hardware support.

5. Parallel programming.

5.1. Basic synchronization and communication mechanisms for distributed memory.

5.2. Strategies for problem decomposition.

5.3. Programming models and their mappings.

5.3. Map-Reduce model case study.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A cadeira visa fornecer a um Engenheiro Informático um conjunto de conhecimentos e competências que lhe permitem dominar a problemática da programação de sistemas com múltiplos processadores, tema de relevância crescente. Para atingir esse objetivo, a cadeira começa por apresentar os conceitos fundamentais da execução concorrente de programas, e por alertar para a importância dos multiprocessadores. Seguem-se as duas partes nucleares:

- A programação em memória partilhada, onde se apresenta um conjunto de questões transversais à programação concorrente, sobretudo relacionadas com a correção de programas.

- A programação de agregados (clusters) de computadores, onde o ênfase é na metodologia de desenvolvimento de algoritmos e programas paralelos.

Na componente prática, imprescindível à consolidação dos conceitos apresentados nas aulas teóricas, a opção pela linguagem Java permite o uso de um conjunto de valiosas ferramentas; adicionalmente, a sua orientação pelos objetos permite encapsular, quando conveniente, detalhes desnecessários.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The course aims to expose students to a set of knowledge and competences to enable them to tackle the increasingly relevant problem of programming systems with multiple processors. To achieve this, the course begins by presenting the fundamental concepts related to the concurrent execution of programs. This is followed by two pivotal parts:

- Shared memory programming, where a series of challenges underlying concurrent programming are presented, mostly related to program correction.

- Cluster programming, where the emphasis is on the methodology for developing parallel algorithms and programs.

In the practical component of the course, which is indispensable for consolidating the concepts taught in lectures, the choice of using Java provides access to a series of valuable tools; additionally, the object-oriented nature of the language allows for encapsulating, when convenient, unnecessary details.

3.3.7. Metodologias de ensino (avaliação incluída):

Semanalmente são lecionadas uma aula teórica com a duração de duas horas e uma aula prática com a duração de duas horas.

Nas aulas teóricas são apresentados os pontos essenciais da matéria que deverão ser complementados com a leitura dos manuais aconselhados. As aulas laboratoriais destinam-se à apresentação e realização dos trabalhos práticos, prevendo-se a realização de dois trabalhos práticos: um relacionado com os pontos 1 a 4 da matéria e outro sobre o ponto 5.

Os alunos têm ao seu dispor uma página da cadeira que mantém informação atualizada sobre o funcionamento da cadeira, sendo disponibilizados acetatos da matéria teórica juntamente com guiões dos trabalhos práticos. A avaliação é contínua e composta por 2 trabalhos práticos de grupo (15% da nota cada), 2 testes (35% da nota final cada) durante o semestre.

3.3.7. Teaching methodologies (including evaluation):

There is one two-hour lecture per week, and one two-hour laboratory class.

In lectures, the key points of the syllabus are presented. This presentation should be complemented with studying the recommended readings. The laboratory classes target the presentation and resolution of the two class projects, one of which covers points 1 through 4 of the syllabus, while the other covers point 5.

Students have at their disposal a course webpage, which maintains up-to-date information on the course, and provides access to the slides used in lectures and to the supporting material for laboratory classes.

The course grade is assessed throughout the semester, and it is computed based on 2 projects solved in a team (each weighing 15% of the final grade) and 2 quizzes (35% each).

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As metodologias de ensino têm como objetivo permitir conhecer as abstrações e ferramentas usadas na programação concorrente e paralela. Tal é conseguido através da uma integração cuidada entre os

conhecimentos explicados na aula teórica e a aprendizagem dos conceitos através de exercícios de programação, quer nas aulas práticas quer nos trabalhos práticos de grupo. Estes dois trabalhos permitem aprofundar os conhecimentos dos dois pilares fundamentais da cadeira: a programação concorrente e a programação paralela.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The goal of the teaching methodologies is to enable students to gain a deep understanding of the tools and abstractions used in the context of parallel and concurrent programming. This is achieved through a careful integration of the material that is conveyed in lecture, and the acquisition of knowledge through programming exercises, both in laboratory classes and in the class projects. The two class projects, solved in teams, allow for gaining an in-depth knowledge of the two pillars of the course, namely concurrent and parallel programming.

3.3.9. Bibliografia principal:

- [1] M. Herlihy, N. Shavit. The Art of Multiprocessor Programming. Morgan Kaufman, 2008.*
- [2] P. Pacheco. An Introduction to Parallel Programming. Morgan Kaufman, 2011*
- [3] T. Mattson, B. Sanders, B. Massingill. Patterns for Parallel Programming. Addison-Wesley, 2004.*

Mapa IV - Construção e Verificação de Software / Software Construction and Verification

3.3.1. Unidade curricular:

Construção e Verificação de Software / Software Construction and Verification

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (T-28h; P-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Ricardo Viegas da Costa Seco (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- Construção de desenvolvimento de software confiável tendo em conta concorrência e segurança*
- integração da verificação com os processos de construção de software*
- Raciocínio baseado na lógica na verificação e análise de sistemas de software*
- Análise estática e de verificação de modelos*
- Princípios e técnicas de teste de software*

Saber Fazer:

- Especificar, verificar e raciocinar sobre correção de programas através de asserções lógicas*
- Definir especificações comportamentais (invariantes, precondições e pós-condições) para implementação de módulos e suas interfaces.*
- Conceber e implementar planos de teste.*
- Usar métodos de programação rigorosos e técnicas de verificação em programas concorrentes*
- Conceber políticas de fluxo de informação e sua verificar com asserções e ferramentas.*
- Desenvolver em grupo uma aplicação de média dimensão, estaticamente verificada e testada com um grau de cobertura razoável.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- Construction of trustworthy software, when concurrency and security are major concerns.*
- Tight integration of verification with the software construction process.*
- Logical reasoning in software verification and analysis.*
- Static analysis and software model-checking techniques*
- Principles and techniques for software testing.*

Application:

- Specification, checking and reasoning about program correctness with logical assertions;*
- Behavioral specifications (invariants, preconditions and post-conditions) for module implementations and their interfaces.*

- *Development and implementation of test plans*
- *Programming methods and verification techniques to enforce safety of concurrent*
- *Specification of information flow policies and its verification using assertions and tools.*
- *Develop, in team work, a statically verified and tested (with reasonable coverage) medium size application.*

3.3.5. Conteúdos programáticos:

1. Verificação de Software

Métodos baseados em Asserções e Lógicas de Hoare, Dijkstra, Owicky-Gries e Separação. Inferência de Asserções. Tipos comportamentais. Análise de fluxo de dados, de controle e de sinonímia; Interpretação Abstrata. Verificação de Modelos. Ferramentas associadas.

2. Teste de Software

Testes baseado em Modelos e Falhas. Seleção e Geração de Testes. Execução Simbólica. Ferramentas associadas.

3. Verificação e Programação Concorrente

Partilha, Confinamento, Posse. Controle de Interferência. Verificação de código com monitores e semáforos usando invariantes de recursos. Controle de concorrência a partir de especificações comportamentais.

4. Verificação e Programação Segura

Segurança de fluxos de informação e não-interferência. Segurança baseada nas linguagens. Modelos de programação para controle de acessos. Certificação de código. Ferramentas associadas.

5. Exercícios de Desenvolvimento e Projeto Final

Uso de ferramentas (JBoss, Verifast, SPIN; Jif)

3.3.5. Syllabus:

1. Software Verification

Assertion methods and Hoare, Dijkstra and Owicky-Gries Logics; Assertion Inference; Behavioral types. Data flow, control flow, and alias analyses; Abstract interpretation; Model-checking. Tools.

2. Software Testing

Model-based testing; Test selection and test generation; Fault-based testing. Symbolic execution; Automated testing. Tools.

3. Concurrent Programming and Verification

Sharing, confinement, ownership. Control of interference. Reasoning about concurrent code with monitors and locks based on resource invariants. Construction of concurrency control code from behavioral specs.

4. Secure Programming and Verification

Information flow security and non-interference. Language based security. Programming models for access control. Certification of code. Tools.

5. Hands On Exercises / Final Project

Sequence of programming challenges, involving tool usage (JBoss, Verifast, SPIN; Jif). Final (team work) project.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Existe uma correspondência evidente entre os conteúdos e os objetivos.

Um princípio pedagógico da UC é a apresentação coesa da teoria com a sua aplicação prática, em geral envolvendo ferramentas.

Os objetivos de “Saber” são cobertos nos tópicos 1, 2, 3 e 4, com ênfase em 1, 2.

Os objetivos de “Saber fazer” são cobertos nos tópicos 1, 2, e 3 e 4 com ênfase em 3, 4.

3.3.6. Demonstration of the syllabus coherence with the curricular unit’s intended learning outcomes:

There is an evident correspondence between the syllabus and the curricular unit’s objectives.

A pedagogical principle enforced in the course is the tightly coupled presentation of theory with application, in general involving the use of tools

The “Knowledge” learning outcomes are covered in items 1, 2, 3 e 4, with an emphasis in 1, 2 .

The “Knowledge Application” learning outcomes are covered in items 1, 2, e 3 and 4, with emphasis in 3,4.

3.3.7. Metodologias de ensino (avaliação incluída):

O ensino está organizado em aulas teóricas e práticas. Existem notas e slides especialmente preparados, que seguem de perto os conteúdos das aulas teóricas.

Nas aulas práticas os alunos discutem e resolvem exercícios propostos pelo docente, de uma lista predefinida. Nas aulas teóricas são apresentados os conceitos e discutidas situações problemáticas. Tipicamente , algumas

das competências de saber fazer são também exercitadas nas aulas teóricas, de forma a aumentar a ligação

entre os conceitos teóricos e a sua aplicação.

A avaliação contínua consiste em dois testes (um intermédio, outro no final do semestre) e 1 projeto de desenvolvimento de software, realizado em grupo. São também apresentados 4 pequenos trabalhos, que deverão ser resolvidos nas aulas laboratoriais (uso de ferramentas). Os alunos que tenham reprovado na avaliação contínua com uma nota igual ou superior a 8 podem tentar fazer o exame final, cuja nota combinada com a dos trabalhos práticos produz a nota final.

3.3.7. Teaching methodologies (including evaluation):

The course is organized in recitation lectures (2T) and laboratory classes (2P). There are written lecture notes and hand out slides, which closely follow the presentation in recitation lectures.

In the laboratory classes students discuss and solve problems proposed by the instructor from a predefined list. In the recitation lectures the instructor presents and motivates concepts and applications are discussed and exemplified. Typically, some “knowledge application” learning outcomes are also exercised in the recitation, so to promote a close connection between the theoretical concepts and their application.

Evaluation consists in midterm and final tests and one software development project, to be done by a team. Any student that fails in the basic evaluations scheme with a grade equal or higher than 8/20 is admitted to the final exam, which combines with the grading of the assignment to produce the final grade.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os trabalhos práticos e o projeto de grupo permitem prender os alunos a um forte ritmo de acompanhamento da matéria. O uso de ferramentas de análise estática e de teste promove a motivação dos alunos para a adoção de técnicas rigorosas de verificação de software, já que se torna muito claro o poder das mesmas no aumento de confiança do software produzido. A formação teórica, que contribui para a compreensão profunda dos mecanismos de funcionamento das ferramentas também reforça os objetivos de aprendizagem. A metodologia seguida, apelando à intuição e esclarecendo bem a utilidade dos resultados teóricos em cada passo, tem dado bons resultados.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit’s intended learning outcomes:

The laboratory work assignments and the team project contribute to bind the students to a strong work rhythm while following the course presentation. The use of static analysis and testing tools facilitates the motivation of students towards the adoption of rigorous verification techniques, since they are able to clearly see the empowering effect on the robustness of the software produced. The study of theory, contributing for the deep understanding of the working principles of tools, also enforces the learning outcomes. The methodology we have devised, which appeals to intuition, and clarifies at each step the usefulness of the presented theoretical results, seems to provide good results.

3.3.9. Bibliografia principal:

Text Book References:

- R. D. Tennent, *Specifying Software: A hands on introduction*, 2002.
- Goetz et al. *Java Concurrency in Practice*, Addison-Wesley, 2006.
- A. Sabelfeld, A. C. Myers, *Language Based Information Flow Security*, 2004.

Additionally, several classical papers by Hoare, Dijkstra, Brinch Hansen, Doug Lea, O’Hearn, Schneider.

Mapa IV - Engenharia de Software / Software Engineering

3.3.1. Unidade curricular:

Engenharia de Software / Software Engineering

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Ana Maria Dinis Moreira (T-28h; P-56h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

N/A

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Princípios, objetivos e conceitos fundamentais da Engenharia de Software (ES).*
- *As problemáticas do desenvolvimento de software em larga escala.*
- *Paradigmas de desenvolvimento.*
- *Técnicas e ferramentas de elicitação de requisitos, análise e desenho.*
- *Padrões de desenho mais comuns.*
- *Estilos básicos de desenho arquitetural.*
- *Reutilização de software.*
- *Verificação e Validação (V&V).*
- *Gestão da qualidade do software.*
- *Evolução do software.*
- *Gestão de projetos.*

Saber fazer:

- *Identificar e especificar os vários tipos de requisitos.*
- *Selecionar alternativas arquiteturais.*
- *Modelar, desenhar e implementar o software com vistas à reutilização e à evolução.*
- *Aplicar as técnicas de V&V.*
- *Gerir um projeto de software, incluindo pessoas, tarefas e custos.*
- *Aplicar as várias técnicas de ES a diversos tipos de problemas.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *The principles, objectives and key concepts of Software Engineering (SE).*
- *The issues of software development on a large scale.*
- *Paradigms of development.*
- *Techniques and tools for requirements elicitation, analysis and design.*
- *Most common design patterns.*
- *Basic styles of architectural design.*
- *Software reuse.*
- *Verification and Validation (V & V).*
- *Management of software quality.*
- *Software Evolution.*
- *Project management.*

Application:

- *Identify and specify the various types of requirements.*
- *Select architectural alternatives.*
- *Model, design and implement software for reuse and evolution.*
- *Apply V & V techniques.*
- *Manage a software Project, including people, tasks and costs.*
- *Apply the various ES techniques to diverse types of problems.*

3.3.5. Conteúdos programáticos:

Princípios e objetivos da ES

Conceitos, modelos de processos, atividades, ética profissional

Modelos de Processos de Negócios

Requisitos

As atividades principais de Engenharia de Requisitos

Classificação de requisitos

Modelos de requisitos

Rastreabilidade

Especificação formal

Desenho de Software

Paradigmas de Desenho

Dos requisitos ao desenho

- Transformações de modelos

Desenho de interface para utilizador

Introdução ao Desenho Arquitetural

- Conceitos básicos de desenho arquitetural

- Principais estilos/padrões arquiteturais

Reutilização de software

- Conceitos básicos de reutilização de software

- *Principais técnicas de reutilização*
- Verificação e Validação*
- Conceitos básicos de V&V*
- Prototipagem*
- Testes*
- Qualidade*
- Processo de medição*
- Qualidades de desenho internas e externas*
- Tradeoffs*
- Evolução*
- Sistemas legados*
- Mudança de software*
- Gestão de configuração*
- Gestão de software*
- Gestão de risco*
- Estimativa de custos*
- Gestão de pessoal*

3.3.5. Syllabus:

- Principles and Goals of SE*
- Concepts, software process models, software activities, professional ethics*
- Business Process Modelling*
- Requirements*
- The main activities of Requirements Engineering*
- Requirements classification*
- Requirements models*
- Requirements traceability*
- Formal specification*
- Software Design*
- Design Paradigms*
- From requirements to design*
- *Model transformations*
- User Design Interface*
- Introduction to architectural design*
- *Basic architectural design concepts*
- *Main architectural styles/patterns*
- Software reuse*
- *Basic software reuse concepts*
- *Main techniques of software reuse*
- Software Verification and Validation*
- V&V concepts*
- Prototyping*
- Testing*
- Software Quality*
- Software process measurements*
- Internal and external design qualities*
- Quality Tradeoffs*
- Software Evolution*
- Legacy systems*
- Software change*
- Configuration management*
- Software Management*
- Risk management*
- Estimation costs*
- Managing people*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Inicialmente os conceitos introdutórios de Engenharia de Software (ES) são apresentados juntamente com os seus processos e atividades. Em particular são apresentados os modelos de processo do negócio. Os conceitos, os modelos e as principais atividades de requisitos são detalhados, onde são apresentadas algumas técnicas específicas. A fim de obter um sistema mais preciso pode-se utilizar a especificação formal dos requisitos antes da fase de desenho. O desenho de software envolve a transição dos requisitos para os modelos de desenho, o desenho arquitetural, e a utilização de técnicas específicas de reuso. De seguida são

discutidos conceitos e técnicas de V&V e de qualidade de processo. Por fim, conceitos e atividades de evolução e gestão de software são apresentados.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

Initially the introductory concepts of Software Engineering (SE) are presented along with its processes and activities. In particular we present the business process modeling. The concepts, models and the main requirements activities are detailed, where some specific techniques are introduced. In order to obtain a more precise system, requirements formal specification can be used. The software design phase involves the transition from requirements to design models, architectural design, and specific techniques for reuse. Then concepts and techniques of V & V and process quality are also discussed. Finally, concepts and activities of software evolution and management are presented.

3.3.7. Metodologias de ensino (avaliação incluída):

No início das aulas os alunos receberão o planeamento para os tópicos de investigação e o trabalho prático, incluindo os pontos que devem ser endereçados. Cada ponto está relacionado com tópicos específicos apresentados nas teóricas. Um prazo é dado para a entrega dos trabalhos.

As aulas teóricas são dadas em sala de aula equipada com um projetor e slides em Powerpoint. As aulas práticas são realizadas em salas equipadas com computadores e projetor.

A avaliação consiste das seguintes componentes obrigatórias: um trabalho prático entregue em duas fases; dois testes. O trabalho é realizado em grupo e os testes, individualmente.

A nota é uma média ponderada do trabalho prático (Fase I = 30%; Fase II = 30%), e da nota dos testes (20% cada um). Acesso ao exame de recurso depende se o aluno obteve um desempenho satisfatório com média entre 9,5 e 20. Os alunos que não cumprirem qualquer uma das partes acima são classificados como "Reprovado".

3.3.7. Teaching methodologies (including evaluation):

In the beginning of the course the students receive the schedule for their research topics and practical work, which includes several questions. Each question relates to specific topics addressed in theory lectures. A time-period is specified for answering the questions.

Lectures are given in a class room equipped with a computer projetor, where each topic is presented with the help of Powerpoint. The practical sessions take place in a class room equipped with computers and projetor. The assessment is divided into the obligatory parts: a practical work delivered in 2 phases; 2 tests. Practical works and essays are done in groups and tests, individually.

The final grade is the weighted average of the grades obtained in the practical work (Phase I = 30%; Phase 2 = 30%), and the tests (20% each). Access to the ressit exam is conditional upon the satisfactory performance with a grade of at least 9,5 out of 20. Students who do not deliver any part depicted above are classified as "Failed".

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas teóricas todos os tópicos do programa são apresentados, sempre ilustrados com sistemas inspirados em situações reais. Nas aulas práticas os conceitos introduzidos na teórica são consolidados com exercícios específicos. Nas práticas também é realizado o acompanhamento dos trabalhos a fim de avaliar o seu progresso.

O trabalho prático serve para os alunos praticarem os conceitos adquiridos de forma integrada, aplicada a um caso de estudo comum. O trabalho prático fomenta o trabalho em equipa, enquanto que os testes avaliam a performance individual.

O trabalho prático e as aulas práticas oferecem oportunidades (através da aplicação a estudos de caso) aos alunos para praticarem técnicas e ferramentas de elicitação de requisitos, análise, desenho arquitetural, desenho detalhado, V&V (e.g. através de prototipagem e inspeções) e reutilização de software (e.g., uso de padrões e componentes). Ênfase também é dada na qualidade dos artefactos que devem ser produzidos (e.g., estar bem modularizados) com vistas a uma evolução mais eficiente. Esses estudos de caso devem também aplicar técnicas essenciais de gestão de projetos (escalonamento de tarefas).

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In the lectures all topics of the program are presented, always illustrated with systems inspired by real situations. In practical classes the concepts introduced in the theory are consolidated with specific exercises. In practical classes it is also carried out monitoring of the practical work and essay to assess their progress.

The practical work is for students to practice the concepts acquired in an integrated way, applied to a common case study. The practical work fosters teamwork, while the tests assess individual performance.

The practical work and practical classes offer opportunities (through application to case studies) for students to practice techniques and tools for requirements elicitation, analysis, architectural design, detailed design, V & V (e.g., through prototyping and inspections) and software reuse (e.g., using patterns and components). Emphasis is also given to the quality of the artifacts to be produced (e.g., well-modularized), aiming at a more efficient evolution. These case studies should also apply essential techniques of project management (task scheduling).

3.3.9. Bibliografia principal:

- *I. Sommerville, Software Engineering, Addison-Wesley, 9th edition, 2010*
- *R.S. Pressman, Software Engineering: A Practitioner's Approach, 7th edition, 2009*
- *A. Lamsweerde, Requirements Engineering, Wiley, 2009*
- *I. Alexander, N. Maiden, Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle, Wiley, 2004*
- *S. Pfleeger, J. Atlee, Software Engineering - Theory and Practice, Prentice Hall, 2005*
- *L. Bass, P. Clements, R. Kazman, "Software Architecture in Practice", 2nd edition, Addison-Wesley, 2003.*
- *P. Clements, F. Bachmann, Le. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, J. Stafford, "Documenting Software Architectures: Views and Beyond", Addison-Wesley, 2003*
- *C. Kaner, J. Falk, H. Q. Nguyen, Testing Computer Software, 2nd Edition, Wiley, 1999*
- *T. Mens, S. Demeyer, Software Evolution, Springer, 2010*

Mapa IV - Interação Pessoa-Máquina / Human-Computer Interaction

3.3.1. Unidade curricular:

Interação Pessoa-Máquina / Human-Computer Interaction

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Nuno Manuel Robalo Correia (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Teresa Isabel Lopes Romão (Regente) (T-14h; P-28h)

João Miguel da Costa Magalhães (T-14h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Ter consciência da importância da interface como componente de um sistema interativo.*
- *Compreender os fatores humanos que condicionam a utilização dos sistemas interativos.*
- *Compreender e utilizar novos paradigmas tecnológicos e metáforas de interação.*
- *Conhecer e aplicar os princípios de usabilidade.*
- *Conhecer e aplicar diversas técnicas de prototipagem.*
- *Conhecer e aplicar diversas técnicas de avaliação de interfaces.*
- *Enquadrar IPM num Projeto de Engenharia.*
- *Discutir a investigação recente no campo da interação pessoa-máquina (HCI)*

Saber fazer:

- *Analisar e especificar os requisitos dos utilizadores.*
- *Desenhar, implementar e avaliar interfaces (Desenho iterativo).*
- *Criar soluções inovadoras de resposta a problemas de interação.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Be aware of the importance of the interface as a component of an interactive system.*
- *Understand the human factors which drive the usage of computer systems.*
- *Understand novel paradigms for human-computer interaction*
- *Know and apply usability principles.*
- *Know and apply prototyping techniques.*
- *Know and apply interfaces evaluation techniques.*

- *Fit HCI in the engineering project.*
- *Describe and discuss current research in the field of HCI.*

Application:

- *Analyse and specify users requirements.*
- *Iterative design: Design, implement and evaluate user interfaces.*
- *Develop creative capabilities to come up with innovative solutions for interaction problems.*

3.3.5. Conteúdos programáticos:

1. *Introdução - Human-Computer Interaction (HCI): o quê, para quê, quando?*
2. *Princípios de usabilidade*
3. *Características dos sistemas interativos*
4. *Fatores humanos e tecnológicos que condicionam a utilização de um sistema interativo*
5. *Desenho centrado no utilizador e processo de desenho iterativo de interfaces*
6. *Análise de tarefas e utilizadores*
7. *Sketching e prototipagem*
8. *Princípios de desenho de interação*
9. *Estilos de interação*
10. *Desenho gráfico*
11. *Paradigmas de interação*
12. *Técnicas de avaliação*
13. *Perspetivas futuras*

3.3.5. Syllabus:

1. *Introduction - Human-Computer Interaction (HCI): What? Why? When?*
2. *Usability principles*
3. *Characteristics of interactive systems*
4. *Human factors in the HCI*
5. *User centered design and iterative design process*
6. *User and task analysis*
7. *Sketching and prototyping*
8. *Interaction design principles*
9. *Interaction styles*
10. *Graphic design*
11. *Interaction paradigms*
12. *Evaluation techniques*
13. *Future perspectives*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O programa inclui o processo e os princípios de desenvolvimento de interfaces para sistemas interativos, focando diversas metodologias para as diferentes fases do processo: análise de requisitos, desenho, desenvolvimento e avaliação. São apresentados os fatores humanos e tecnológicos que condicionam a utilização de um sistema interativo e que determinam os princípios de usabilidade. É discutida a investigação recente no campo da interação pessoa-máquina (HCI) e analisadas as perspetivas futuras.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The program includes the process and principles of user interfaces development, focusing on different methodologies for the different iterative design process stages: user and task analysis, design, development and evaluation. Human factor and technological principles, which determine the use of interactive systems, are presented and the principles of usability are studied. Recent research in the field of human-computer interaction (HCI) is discussed and future perspectives are analysed.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas é apresentada a matéria, ilustrada com exemplos de aplicação. As aulas laboratoriais destinam-se à apresentação e realização dos trabalhos práticos sobre a matéria apresentada na aulas teóricas.

A avaliação continua da unidade curricular é composta por 4 elementos: 2 testes escritos individuais efetuados ao longo do semestre e 2 trabalhos práticos que no seu conjunto correspondem a um projeto de desenvolvimento de uma interface para um sistema interativo.

Fórmula de cálculo da nota final:

NotaFinal = 35%Teste1 + 35%Teste2 + 15%TrabPratico1 + 15%TrabPratico2

A aprovação à unidade curricular exige as seguintes notas mínimas:

- *(média (Teste1; Teste2) >= 10) AND*
- *(Trab_Pratico1 >= 10) AND*
- *(Trab_Pratico2 >= 10) AND*
- *(Teste1 >= 8) AND*
- *(Teste2 >= 8)*

Os alunos que obtiveram notas positivas nos trabalhos práticos e não obtiveram aprovação nos testes escritos poderão ainda realizar um exame, cuja nota substituirá a nota dos testes no cálculo da nota final.

3.3.7. Teaching methodologies (including evaluation):

In the theoretical lectures the course content is presented, illustrated with application examples. The lab classes are intended for presentation and implementation of the practical work regarding the content exposed during theoretical lectures.

The evaluation of the course consists of four elements: two mid-term written individual tests and two practical works, which together account for a project to develop the user interface for an interactive system.

Final grade formula:

Final_grade = 35%Test1 + 35%Test2 + 15%Prat_Work1 + 15%Prat_Work2

Course approval requires the following minimal grades:

- *(mean (Test1; Test2) >= 10) AND*
- *(Prat_Work1 >= 10) AND*
- *(Prat_Work2 >= 10) AND*
- *(Test1 >= 8) AND*
- *(Test2 >= 8)*

Students approved in the practical works and not approved in the tests may also perform a final exam, which grade replaces the grades of the tests in the final grade formula.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas teóricas são expostos os fundamentos teóricos da unidade curricular e exemplificada a sua aplicação através de exemplos.

Para além das bibliografia recomendada, os alunos têm acesso aos slides apresentados nas aulas teóricas e a artigos científicos acerca dos assuntos tratados, de forma a ajudar a sua compreensão e enquadramento na área de Interação Pessoa-Máquina.

Durante as aulas práticas, os alunos têm que trabalhar em grupo e aplicar os conhecimentos proporcionados pelas aulas teóricas e pelos materiais didáticos no desenvolvimento iterativo de um projeto de uma interface para um sistema interativo. Este projeto pretende, na medida do possível, simular o trabalho a realizar por uma equipa de trabalho real no desenvolvimento da interface para um sistema interativo, encadeando as várias fases que o compõem e experimentando as metodologias aprendidas.

Algumas aulas práticas específicas são de presença obrigatória, pois promovem o trabalho entre equipas ou correspondem à avaliação do trabalho.

Os testes escritos individuais permitem a avaliação mais precisa dos conhecimentos adquiridos por cada aluno.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

During theoretical classes, the theoretical fundamentals of the course are presented and its application illustrated through examples.

In addition to the recommended readings, students have access to the slides presented during the lectures and additional scientific papers regarding the addressed topics, in order to facilitate their understanding and framing into the area of Human-Computer Interaction.

During the lab classes, students have to work together and apply the knowledge provided by both the lectures and educational materials for the iterative development of an interface for an interactive system. This project aims at simulating, as far as possible, the work to be undertaken by a real work team in developing the interface for an interactive system, chaining the different project stages and experiencing the methodologies learned. Some specific lab classes are mandatory, because they promote teamwork or correspond to evaluation procedures.

The individual written tests allow more accurate assessment of knowledge acquired by each student.

3.3.9. Bibliografia principal:

Recommended

- *Dix, A., Finlay, J., Abowd, G. and Beale, R., Human-Computer Interaction. Prentice Hall Europe, London, 2003.*
- *Norman, D., The Design of Everyday Things. MIT Press, 1998.*
- *Nielsen, J., Usability Engineering, Academic Press, 1993.*

Complementary

- *Mullet, K. and Sano, D., Designing Visual Interfaces, Prentice Hall, 1995*
- *Shneiderman, B., Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, Reading, Massachusetts, 1998.*
- *Moggridge, B., Designing Interactions. MIT Press, Massachusetts, 2007.*
- *Tufte, E., Envisioning Information, Cheshire, Connecticut Graphic Press, 2003.*

Additional readings will be provided during classes and through the course website.

Mapa IV - Interpretação e Compilação / Interpretation and Compilation

3.3.1. Unidade curricular:

Interpretação e Compilação / Interpretation and Compilation

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Ricardo Viegas da Costa Seco (Regente) (T-28h; P-56h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Esta unidade curricular tem o objetivo de transmitir conhecimentos sólidos na conceção e implementação de linguagens de programação, através do estudo de conceitos sintáticos, semânticos e pragmáticos fundamentais.

Saber

- 1. Conhecer a arquitetura e técnicas usadas no desenho de interpretadores, compiladores e sistemas de tipos*
- 2. Conhecer os elementos base das linguagens de programação e a sua semântica*
- 3. Definir linguagens de programação por composição de elementos fundamentais*

Saber Fazer

- 4. Representar e manipular a sintaxe abstrata de uma linguagem como dados de um programa transformador*
- 5. Descrever a semântica de linguagens através de interpretadores, compiladores e verificadores de tipos*
- 6. Conceber e implementar compiladores para uma máquina virtual industrial*

Competências complementares

- 7. Raciocinar sobre sistemas complexos a vários níveis de abstração*
- 8. Propor e concretizar soluções gerais com base em princípios*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This course aims at teaching solid knowledge and concepts in the design and implementation of programming languages. This goal is reached by the systematic study of the syntax, semantics and pragmatics of programming language.

Knowledge

- 1. To know the architecture and techniques used in the design and implementation of interpreters, compilers and type systems*
- 2. To know the essential components of the design of programming languages and corresponding semantics*
- 3. To define programming languages by composition of base elements*

Application

- 4. To define algorithms of the abstract representation of programs*
- 5. To describe language semantics by interpreting, compiler and verification algorithms*
- 6. To design and implement compiler procedures targeting concrete virtual machines*

Soft-Skills

- 7. To reason about complex systems at different levels of abstraction*
- 8. To design general purpose designs based on first principles*

3.3.5. Conteúdos programáticos:

A.Princípios

- 1.Sintaxe das Linguagens de Programação
- 2.Semântica das Linguagens de Programação
- 3.Níveis de Interpretação (Interpretação, compilação e sistemas de tipos)

B.Interpretação de Programas

- 1.Valores e Expressões: Definições e Ambiente
- 2.Estado e Referências: Modelo ambiente-memória
- 3.Abstração funcional: funções de primeira classe e ordem superior
- 4.Definições recursivas e ambientes circulares
- 5.Valores estruturados: registos e valores recursivos
- 6.Objetos e classes: representação de objetos usando registos e fechos

C.Sistemas de Tipos

- 1.Princípios, objetivos e limitações da análise estática
- 2.Sistemas de tipos simples
- 3.Algoritmos de verificação e inferência de tipos

D.Compilação de Programas

- 1.Arquitetura de um compilador
- 2.Ambientes de suporte à execução (máquinas de registos, pilha de avaliação, pilha de chamada)
- 3.Tradução de código dirigida pela sintaxe
- 4.Geração de código para uma máquina virtual industrial (CLR,LLVM)

3.3.5. Syllabus:

A.Principles

- 1.Programming Language Syntax
- 2.Programming Language Semantics
- 3.Interpretation levels

B.Program interpretation

- 1.Expressions and values; Definitions and environments
- 2.State and references; memory-environment model
- 3.Functional abstraction; first-class and higher order functions
- 4.Recursive definitions
- 5.Structured values: records and recursive values
- 6.Objects and classes: Objects represented with records and closures

C.Type Systems

- 1.Principles, goals and limitations of static analysis methods
- 2.Simple type systems: functional types
- 3.Verification and inference algorithms

D.Compilers and Runtime support systems

- 1.Compiler architecture
- 2.Runtime support systems
- 3.Syntax driven code transformation
- 4.Code generation to intermediate languages (CLR, LLVM)

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Esta unidade tem por objetivo transmitir um conjunto de princípios(A) abordados transversalmente, e que contribuem para atingir os objetivos 1,4 e 8. A definição de algoritmos interpretadores de forma incremental(B.1-6) é o fio condutor para atingir os objetivos 2, 3 e 5, e permite definir com precisão a semântica das construções nos vários níveis de interpretação(B,C,D) (semântica estática e semântica dinâmica) (objetivo 7). É colocada ênfase na composicionalidade dos elementos base(B), promovendo a compreensão de construções de análise mais complexa, comuns nas linguagens de programação de uso corrente (objetivo 8). Por outro lado, a exigência de construção de um projeto de média dimensão, promove o conhecimento profundo acerca da arquitetura de um interpretador (B), de um compilador(C.3), de um sistema de suporte à execução (D.2) assim cumprindo os objetivos 3,4 e 6, bem como qual o papel de um verificador baseado em sistemas de tipos, objetivo 5.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course aims at transmitting a set of cross cutting principles(A) that contribute to reach goals 1, 4, and 8. The incremental definition of interpreter algorithms(B.1-6) constitutes a central line to reach goals 2,3,and 5, allowing to precisely define the semantics of the different language elements at different interpretation levels (B,C,D) - static semantics and dynamic semantics (goal 7). We focus on the compositionality of base

elements(B) as a way of understanding more complex constructs which are common in mainstream programming languages (goal 8). On the other hand, the development of a middle-sized project promotes the deeper understanding of the architecture of an interpreter (B), compiler (C.3), or a runtime support system (D.2) thus contributing to goals 3,4, and 6, and the role of a typechecker algorithm, goal 5.

3.3.7. Metodologias de ensino (avaliação incluída):

A metodologia pedagógica proposta para esta unidade curricular centra-se na construção gradual de uma linguagem de programação. O ponto de partida é uma linguagem núcleo de expressões aritméticas, à qual se vão acrescentando novos elementos fundamentais numa espiral até se obter uma linguagem com funções de primeira classe e de ordem superior, capazes de exprimir as construções mais sofisticadas das linguagens de programação como classes, mixins ou closures.

A avaliação decompõe-se em dois testes contribuindo cada um 30% para a nota final, o primeiro focado em aspetos básicos de definição de semântica de linguagens, e o segundo, na verificação através de sistemas de tipos. O restante da nota final (40%) é obtida através do desenvolvimento de um projeto de construção pragmática de um interpretador, compilador e sistema de tipos para uma linguagem dada.

3.3.7. Teaching methodologies (including evaluation):

The teaching methodologies prescribed for this course is focused in the incremental construction of a programming language. The base language is a core expression language containing only arithmetic operators. The course follows in a spiral of new fundamental elements added to the language. The target language is a programming language with first-class and high order functions, capable of expressing more sophisticated constructs as classes, mixins, or closures.

The evaluation comprises two tests, valued 30% in the final grade. The first test focuses on basic aspects of programming language semantics, the second is focused on type systems. The remainder of the final grade is obtained by developing a medium-sized project targeting a pragmatic programming language. The project comprises an interpreter, compiler and type system.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A abordagem proposta como metodologia pedagógica permite a definição com precisão da semântica de cada elemento fundamental das linguagens de programação (objetivo 2). A definição da semântica das linguagens é feita incrementalmente e a par aos vários níveis de interpretação (dinâmico e estático) tal como projetado pelo objetivos 1 e 3. Por um lado, as intuições e definições formais mais profundas são expostas em aula teóricas, por outro lado, consolidam-se os conhecimento transmitidos através do exercício, em aula prática de laboratório, de construção compassada de um crescendo de interpretadores, compiladores e sistemas de tipos segundo a espiral desenhada pelas aulas teóricas (objetivos 4 e 6). O processo de aprendizagem culmina na construção de um projeto de dimensão significativa, que abarca aspetos de composição de elementos da linguagem não triviais (objetivos 7 e 8).

Um dos conceitos mais importantes subjacentes ao desenho de linguagens de programação é o conceito de abstração presente aos mais diversos níveis (funcional, de dados, de tipos). No decorrer das aulas, quer teóricas, quer práticas, os alunos são expostos a situações em que se exige a definição de semânticas a diferentes níveis de abstração (objetivos 3 e 7). Por exemplo, definir a semântica operacional de uma classe ou objeto por tradução noutras construções, e definir a semântica de tipos de forma primitiva.

Outra parte importante dos ensinamentos da unidade curricular prende-se com o conhecimento profundo dos modelos de computação usuais (objetivos 1 e 6). Esse conhecimento é adquirido no desenvolvimento de funções de transformação de compilação para linguagens intermédias, focando em máquinas virtuais reais como a CLR ou a LLVM (objetivos 1 e 6).

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Our methodological approach allows the precise definition of the semantics of each one of the programming language base elements which corresponds to goal 2. The semantics of programming languages is incrementally introduced at different interpretation levels (dynamic and static) as predicted by goals 1 and 3. On the one hand, all intuitions and formal definitions are presented during lectures, on the other hand, this knowledge is consolidated by lab exercises targeting the step-by-step construction of small interpreters, compilers, and type systems (goals 4 and 6). The learning process ends with the construction of a medium-sized project comprising non-trivial composition of programming language base elements (goals 7 and 8).

One of the main underlying aspects of programming language design is the concept of abstraction applied at different levels (functional, data, type). During lectures and lab classes, students are compelled to act on situations that demand the definition of semantics at different levels (goals 3 and 7). For instance, to define the operational semantics by translation in the case classes and objects, and define their type semantics using the class and object constructs.

Another relevant part of the syllabus refers to the deep knowledge of the supporting computation models (goals 1 and 6). This topic is explained throughout several lectures and applied by developing several source to source transformations targeting intermediate languages, focused on real virtual machines like the CLR or LLVM (goals 1 and 6).

3.3.9. Bibliografia principal:

- *“Concepts in Programming Languages”, John C. Mitchell, Cambridge University Press. ISBN 0 521 78098 5*
- *“Essentials of Programming Languages”, Daniel Friedman, Mitchell Wand, Christopher Haynes, MIT Press.*
- *“Compiling for the .Net Common Language Runtime (Clr) (.NET Series)”, John Gough, Prentice Hall PTR*
- *“Modern Compiler Implementation in Java” Andrew W. Appel, Cambridge University Press*
- *“The Study of Programming Languages”, Ryan Stansifer, Prentice Hall International Edition.*

Mapa IV - Representação de Conhecimento e Raciocínio / Knowledge Representation and Reasoning

3.3.1. Unidade curricular:

Representação de Conhecimento e Raciocínio / Knowledge Representation and Reasoning

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Júlio Alves Alferes (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Alexandre Carvalho Pinheiro Leite (Responsável e Regente) (28T; 56P)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Principais formalismos baseados em lógica para representação e raciocínio sobre conhecimento de senso-comum.*
- *Formalismos gerais para representação e raciocínio sobre Ontologias.*
- *Linguagens para representação e raciocínio sobre Ações*
- *Linguagens para representação de Argumentos e raciocínio Dialectico.*

Saber Fazer:

- *Desenvolver meta-interpretadores simples, em Prolog, para semânticas de programas em lógica não monotónicos*
- *Escolher e empregar os formalismos de representação de conhecimento adequados à representação de conhecimento em áreas de aplicação práticas.*
- *Usar Resolvedores de ASP, XSB-Prolog e DL Reasoners.*

Competências Complementares:

- *Aplicação de conhecimentos teóricos formais em aplicações práticas.*
- *Estar ciente de trade-offs no desenho.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Main logic-based formalisms for common-sense knowledge representation and reasoning*
- *General formalisms for representing and reasoning with Ontologies*
- *Languages for representing and reasoning about actions*
- *Languages for representing arguments and dialectical reasoning*

Application:

- *Develop simple meta-interpreters for semantics of non-monotonic logic programming in Prolog*
- *Choose and employ adequate knowledge representation formalisms to represent knowledge in practical applicaiton areas.*
- *Use ASP Solvers, XSB Prolog and DL Reasoners.*

Soft-skills:

- Apply formal theoretical knowledge in practical applications
- Become aware of design trade-offs

3.3.5. Conteúdos programáticos:

1. *Introdução à Representação de Conhecimento e Raciocínio*
2. *Introdução às Lógicas não-monotónicas*
 - 2.1. *Pressuposto do Mundo Fechado*
 - 2.2. *Circunscrição*
 - 2.3. *Lógica de Default*
 - 2.4. *Logica Autoepistémica*
3. *Programação em Lógica Não-Monotónica*
 - 3.1. *Negação por falha*
 - 3.2. *Semântica dos Modelos Estáveis/Programação por Conjuntos de Resposta*
 - 3.3. *Semântica Bem-Fundada*
 - 3.4. *Semânticas Paraconsistentes*
4. *Logicas de Descrição*
 - 4.1. *A Lógica de Descrição ALC*
 - 4.2. *Famílias de Lógicas de Descrição*
 - 4.3. *Raciocínio com Lógicas de Descrição*
5. *Raciocínio sobre Ações*
 - 5.1. *Cálculo de Situações*
 - 5.2. *Linguagens de Ações*
6. *Teoria da Argumentação*
 - 6.1. *Argumentação Abstrata*
 - 6.2. *Argumentação Estruturada*

3.3.5. Syllabus:

1. *Introduction to Knowledge Representation and Reasoning*
2. *Introduction to Non-monotonic Logics*
 - 2.1. *Closed World Assumption*
 - 2.2. *Circumscription*
 - 2.3. *Default Logic*
 - 2.4. *Autoepistemic Logic*
3. *Non-monotonic Logic Programming*
 - 3.1. *Negation as Failure*
 - 3.2. *Stable-Models Semantics/Answer-Set Programming*
 - 3.3. *Well-founded Semantics*
 - 3.4. *Paraconsistent Semantics*
4. *Description Logics*
 - 4.1. *The Description Logic ALC*
 - 4.2. *Families of Description Logics*
 - 4.3. *Reasoning with Description Logics*
5. *Reasoning about Actions*
 - 5.1. *Situation Calculus*
 - 5.2. *Action Languages*
6. *Theory of Argumentation*
 - 6.1. *Abstract Argumentation*
 - 6.2. *Structured Argumentation*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos programáticos seguem de perto os objetivos de conhecimento estabelecidos, sendo assim clara a coerência entre ambos.

As aptidões objetivo da unidade curricular decorrem da exercitação do programa em atividades práticas laboratoriais.

Os conteúdos programáticos relativos à programação em lógica não monotónica, aprofundados nas aulas práticas através do desenvolvimento de meta-interpretores em Prolog, são coerentes com o objetivo correspondente.

Todo o conteúdo programático contribui para o objetivo relacionado com a escolha e utilização de formalismos adequados à representação de conhecimento em áreas de aplicação práticas.

Para o objetivo relacionado com a utilização de resolvedores de ASP, XSB-Prolog e DL Reasoners, contribuem, diretamente, os conteúdos programáticos relacionados com a programação em lógica não-monotónica (ASP e XSB-Prolog) e Lógicas de Descrição (DL Reasoners), aprofundados nas aulas práticas através da sua utilização

em pequenos trabalhos práticos, e, indiretamente, os conteúdos programáticos relacionados com Raciocínio sobre Ações e Teoria da Argumentação (ASP e XSB-Prolog), pelas relações que existem entre ambos. Todas as competências são também fomentadas pela forma de avaliação da unidade curricular, quer pelos testes/exame quer pelo trabalho prático.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus follows very closely the knowledge learning outcomes established for the course, so it is clear the coherence between the two.

The know-how learning outcomes stem from the exercise of the program in lab sessions.

The syllabus items corresponding to the non-monotonic logic programming, deepened in lab sessions through the development of meta-interpreters in Prolog, are coherent with the corresponding objective.

The entire syllabus contributes to the objective related to the choice and employment of adequate knowledge representation formalisms to represent knowledge in practical application areas.

For the objective related to the use of ASP Solvers, XSB Prolog and DL Reasoners, contribute, directly, the syllabus items related to non-monotonic logic programming (ASP and XSB-Prolog) and Description Logics (DL Reasoners), deepened in lab sessions through their use in small projects, and, indirectly, the syllabus items related to Reasoning about Actions and Argumentation Theory (ASp and XSB-Prolog), given the relationships between both.

All learning outcomes are promoted through the evaluation mechanisms, both by the tests/exam and by the projects.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas e práticas. Nas primeiras são lecionados os conceitos e técnicas relevantes.

Nas aulas práticas são resolvidos problemas, feitas experiências em laboratório de computadores, e (parcialmente) desenvolvidos os trabalhos práticos.

A avaliação de conhecimentos inclui:

- 1 teste individual teórico (ou um exame de recurso), onde é avaliado o conhecimento que os alunos adquiriram dos conceitos e características das técnicas lecionadas.*
- 3 trabalhos práticos, elaborados em grupo de 2 estudantes, que consistem: a) no desenvolvimento de um meta-interpretador em Prolog; b) na modelação de um problema usando Programação por Conjuntos de Resposta e c) na modelação de um problema usando o XSB-Prolog com chamadas a um Resolvedor ASP ou a um DL Reasoner.*

3.3.7. Teaching methodologies (including evaluation):

The syllabus is taught in theoretical and laboratory classes. In the former, the main concepts and techniques are addressed.

The laboratory classes are dedicated to solving problems, experimenting the various concepts, and (partially) developing the project.

The evaluation includes:

- 1 individual test (or an exam) where the knowledge acquired by the students on the concepts and techniques is assessed.*
- 3 practical project, developed in groups of 2 students, consisting of a) the development of a meta-interpreter in Prolog; b) the modelling of a problem using Answer-Set Programming and c) the modelling of a problem using XSB-Prolog with calls to an ASP Solver or a DL Reasoner.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular identificam a) a aquisição de conhecimento sobre técnicas e mecanismos de representação de conhecimento e raciocínio; b) a aplicação prática dos conhecimentos adquiridos e utilização de ferramentas existentes.

O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição de um conjunto de formalismos de representação de conhecimento e raciocínio adequados para lidar com conhecimento de senso-comum, ontologias, ações e diálogos.

O objetivo b) é fundamentalmente atingido nas aulas práticas através da aprendizagem e utilização das ferramentas de raciocínio existentes, seja para experimentar os vários conceitos aprendidos, seja para os aplicar na modelação e implementação dos vários trabalhos.

As matérias relativas ao objetivo a) são avaliadas no teste individual. As matérias relativas ao objetivo b) são

avaliadas nos trabalhos práticos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The objectives of the course identify a) the acquisition of knowledge about techniques and mechanisms for knowledge representation and reasoning; b) the practical application of acquired knowledge, and usage of existing tools.

Objective a) is mainly achieved in lectures with the presentation of a set of formalisms for knowledge representation and reasoning appropriate to deal with common-sense knowledge, ontologies, actions and dialogues.

Objective b) is mainly achieved in practical classes through the learning and usage of existing reasoning tools, be it to experiment with the various learnt concepts, be it to apply them in modelling and implementation of the various projects.

Matters relating to the objectives a) are evaluated through individual tests. Matters relating to objective b) are evaluated through the practical projects.

3.3.9. Bibliografia principal:

Livros de Texto / Text Books

- *Knowledge Representation and Reasoning by Ronald Brachman & Hector Levesque, Morgan Kaufmann 2004.*
- *Handbook of Knowledge Representation edited by Frank van Harmelen, Vladimir Lifschitz and Bruce Porter, Elsevier 2007.*
- *Nonmonotonic Reasoning. Grigoris Antoniou. MIT Press, 1996.*
- *Knowledge Representation, Reasoning and Declarative Problem Solving. Chitta Baral. Cambridge University Press, 2003*
- *The Description Logic Handbook: Theory, Implementation, and Applications. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider. Cambridge University Press, 2003*

Mapa IV - Segurança de Sistemas Computacionais / Computing Systems Security

3.3.1. Unidade curricular:

Segurança de Sistemas Computacionais / Computing Systems Security

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Augusto Legatheaux Martins (T-14h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Henrique João Lopes Domingos (T-14h; P-56h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

A unidade curricular visa a aquisição de conhecimentos, competências e aptidões necessárias para lidar com diferentes facetas da segurança informática em projeto, desenvolvimento, operação e produção de sistemas computacionais seguros.

Saber:

- *Ferramentas criptográficas*
- *Canais seguros de comunicação e segurança de aplicações distribuídas*
- *Autenticação e controlo de acessos*
- *Segurança ao nível do sistema de operação*
- *Segurança ao nível do Software*

Saber Fazer:

- *Utilização dos frameworks de segurança em Java para desenvolvimento de aplicações distribuídas seguras*
- *Contacto com versões seguras de sistemas de operação e ferramentas de deteção de vulnerabilidades e auditoria das mesmas. Gestão de chaves e autenticação com base em hardware dedicado*
- *Utilização de ferramentas de análise estática e dinâmica de código. Análise da execução de código não certificado em ambientes virtuais protegidos. Análise de vulnerabilidades em aplicações WEB.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The course aims at the acquisition of the knowledge and the know-how required to deal with the several facets of security in software projects, in systems development, and while managing and operating secure computing systems.

Knowledge:

- *Cryptographic tools*
- *Secure communications channels and security of distributed applications*
- *Authentication and access control*
- *Operating system security*
- *Software security*

Application:

- *Usage of the security frameworks of Java to develop secure distributed applications*
- *Secure operating systems versions and tools for vulnerabilities detection and auditing*
- *Key management and authentication supported by special hardware*
- *Tools to execute non-trusted software in sand-boxing environments.*
- *Tools for the detection of vulnerabilities in WEB applications and contrameasures.*

3.3.5. Conteúdos programáticos:

1. *Introdução à segurança de sistemas computacionais.*
2. *Criptografia. Cifras simétricas e assimétricas. Sínteses de segurança. Assinaturas digitais. Autenticação e distribuição de chaves. Armazenamento seguro de dados.*
3. *Protocolos e aplicações de segurança em redes. Autenticação X509, PKI e sistema Kerberos. Protocolos seguros: SSL, TLS e HTTPS; PGP, SSH e IPSec. Segurança de serviços WEB.*
4. *Segurança do sistemas de operação. Ameaças, auditoria, conformidade e deteção de intrusões. Máquinas virtuais e técnicas de virtualização. Segurança do Hardware.*
5. *Autenticação e controlo de acessos. Autenticação de utilizadores e sistemas. Modelos de controlo de acessos em sistemas operativos e sistemas distribuídos.*
6. *Execução segura de código. Máquinas virtuais e interpretadores. Código móvel e isolamento de execução.*

3.3.5. Syllabus:

1. *Overview of computer systems security.*
2. *Cryptographic tools. Symmetric and public-key encryption. Secure hash-functions. Digital signatures. Authentication and key-distribution schemes. Protection of stored data.*
3. *Secure network protocols and applications. Authentication services: X509, PKI and Kerberos. Security protocols and applications: SSL, TLS and HTTPS, SSH, PGP. IPSec. Secure Web-Services.*
4. *Operating systems security. Threats, auditing, compliance and intrusion-detection. Virtual machines and virtualization techniques. Hardware security.*
5. *Authentication and access control. Authentication methods for user and systems. Principles and models of access-control in operating system and distributed systems.*
6. *Safe software execution. Virtual machines and their interpreters. Mobile code and sand-boxing techniques.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A unidade curricular destina-se a estudantes do 3.º ou 4.ºs anos do mestrado integrado, e tem por objetivo a aquisição de conhecimentos e competências sobre segurança de redes, segurança dos sistemas de operação e segurança do software e das aplicações.

A segurança de sistemas computacionais está a tornar-se muito importante. O assunto cruza diversas áreas da engenharia informática e envolve também aspetos organizacionais, sociais, éticos e humanos. Trata-se de uma primeira unidade curricular opcional de introdução ao tema que os estudantes são encorajados a seguir. Infelizmente, numa unidade curricular semestral de 6 ECTS é impossível fornecer uma panorâmica absolutamente completa de todas as facetas a considerar.

Para esta unidade curricular foram selecionados um conjunto de tópicos fundamentais: ferramentas criptográficas, canais seguros e autenticação, segurança do sistema de operação, controlo de acessos, segurança do software e segurança na execução de código não certificado.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The course aims to prepare advanced under graduate or graduate students with knowledge, understanding and practical skills in network security, systems security, software security and application-level security.

Security of computational systems is becoming of paramount importance. This subject crosses many areas of computer science and also comprises organizational, social, ethical and human factors. This is a first elective course on the subject that most students are encouraged to follow. Unfortunately, it is impossible in a semester long course to provide a complete survey of all the facets of the subject.

The curricular unit addresses some fundamental topics of the field: cryptography tools, secure channels and

secure authentication, operating system security, software security and controlled execution of non-trusted code.

3.3.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas são usadas para exposição dos temas do programa, centrando-se nos aspetos fundamentais dos tópicos cobertos. As aulas práticas decorrem em laboratório, em sessões programadas de exercícios de programação, demonstrações e avaliação experimental e focam-se no estabelecimento de competências de programação de protocolos e aplicações com base em requisitos e critérios de avaliação de segurança. O desenvolvimento de trabalhos e projetos de avaliação, consolidam os conhecimentos práticos de desenvolvimento de software para sistemas distribuídos, incorporando competências associadas a diversas dimensões da segurança desses sistemas.

Os objetivos são alcançados com o desenvolvimento complementar de trabalhos e mini-projetos de avaliação, exigindo uma participação ativa dos alunos na sua execução.

Componentes de avaliação:

- 1. Dois testes de frequência (50 % da avaliação)*
- 2. trabalhos de avaliação com relatório (25% da avaliação)*
- 3. projeto final em grupo (25% de avaliação).*

3.3.7. Teaching methodologies (including evaluation):

Lectures are intended to support the instructor's presentation of fundamental issues. Practical hands on laboratory sessions (including programming exercises, demonstrations and experimental assessment) are used to train programming techniques and to improve practical skills. Work-assignments and mini-projects complement the required student work.

Assessment Components:

- 1. Two intermediate tests (50 % of the final grade)*
- 2. Written and lab assignments (25% of the final grade)*
- 3. Small group project (25% of final grade).*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os tópicos do programa e a sua sequência, têm em vista estabelecer as valências associadas à sensibilidade e compreensão da problemática da segurança de sistemas computacionais, tendo em vista as suas diferentes facetas, de modo a preparar os alunos para a sua futura intervenção em equipas de desenvolvimento de software, bem como gestão e operação de sistemas computacionais e suas aplicações.

O programa cobre diferentes blocos de conhecimentos que se complementam para uma avaliação, conceção, desenvolvimento e manutenção integrada das propriedades e serviços de segurança de um sistema computacional. Nesta integração conjugam-se desde os conhecimentos fundamentais em técnicas e métodos da criptografia computacional até à conceção e suporte de propriedades e serviços de segurança das comunicações, segurança do sistema operativo e segurança do software.

A unidade curricular propicia o estabelecimento das competências em diferentes domínios específicos da segurança de um sistema computacional, nomeadamente: ao nível da compreensão, desenvolvimento e avaliação experimental de protocolos de sistemas e redes de computadores, ao nível do estabelecimento e identificação de bases confiáveis de computação, ao nível da conceção e implementação de mecanismos e serviços de controlo de acessos em sistemas operativos e aplicações distribuídas, ao nível do conhecimento das técnicas de virtualização e suporte de máquinas virtuais em sistemas operativos seguros, o domínio de métodos e sistemas de autenticação de utilizadores e sistemas de computadores, bem como a aquisição dos fundamentos, técnicas e ferramentas para produção de código seguro e sua execução em segurança.

As aulas práticas em ambiente laboratorial complementam os conhecimentos teóricos com a experiência de desenvolvimento, utilização e avaliação experimental de métodos e ferramentas de desenvolvimento.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The selected topics as well as their sequence is specially fitted to the goal of introducing security in computing systems for Computer Science students that will have future roles in computing systems setup and management as well as in project and software development.

It covers the fundamental building blocks of security: cryptographic tools, authentication methods and protocols, secure channels, trusted computing bases, including hardware and software and the maintenance of their integrity, access-control methods, production of safe software and safe execution of non-certified software.

Laboratory sessions and home assignments complement the conceptual understanding of the above building blocks with acquaintance to the practical methods and tools supporting the installation, management and development of secure computing systems.

3.3.9. Bibliografia principal:

Course text reference:

• *William Stallings and Lawrie Brown, Computer Security - Principles and Practice, Prentice Hall, 2009*

Other reference books:

• *G. Coulouris, J. Dollimore, T. Kindberg and G. Blair. Distributed Systems Concepts and Design, 5th Edition, Addison Wesley, 2011*

• *André Zúquete. Segurança em Redes Informáticas, FCA, 2ª Ed., 2008*

• *M. Correia, P. Sousa, Segurança no Software. FCA Ed., 2010*

• *M. Howard, D. LeBlanc, Writing Secure Code, Microsoft Press, 2001*

• *Scott Oaks, Java Security, 2nd Edition, O'Reilly, 2001*

Mapa IV - Sistemas de Bases de Dados / Database Systems

3.3.1. Unidade curricular:

Sistemas de Bases de Dados / Database Systems

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Júlio Alves Alferes (T-28h; P-56h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

N/A

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

• *Conceitos fundamentais para o funcionamento de um sistema de bases de dados.*

• *Noções de indexação e hashing em bases de dados.*

• *Processamento e otimização de perguntas em bases de dados relacionais.*

• *Concorrência e processamento de transações em bases de dados.*

• *Noções básicas de bases de dados distribuídas.*

Saber Fazer:

• *Otimizar bases de dados para utilizações específicas.*

• *Otimizar o processamento de perguntas, e utilizar as ferramentas comuns em SGBD para esse propósito.*

• *Usar os mecanismo de controlo de transações em bases de dados.*

• *Usar um SGBD para implementação de uma base de dados distribuída*

• *Utilização do SQL e linguagens procedimentais com SQL.*

Competências Complementares:

• *Capacidade de trabalho em equipa.*

• *Capacidade de avaliar soluções com base em resultados experimentais.*

• *Capacidade de avaliação crítica.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

• *Fundamental concepts in database management systems.*

• *Indexing and hashing in databases*

• *Query processing and optimization in relational databases.*

• *Concurrency and transaction processing in databases.*

• *Basic notions of distributed databases.*

Application:

• *Optimize databases for specific uses.*

• *Optimize the processing of queries, using common tools for this of DBMSs.*

• *Use of control mechanisms for concurrency*

• *Use of DBMSs for implementing distributed databases.*

• *Advanced use of SQL and associated procedural languages.*

Soft-skills:

• *Team work.*

• *Capacity to evaluate a solution based on experimental results.*

• *Critical evaluation.*

3.3.5. Conteúdos programáticos:

1.Implementação de SGBDs

1.1.Armazenamento e estrutura de ficheiros

1.2.Indexação

1.3.Hashing estático e dinâmico

2.Processamento e otimização de perguntas em SQL.

2.1.Processamento: esquema geral; algoritmos para operadores de álgebra relacional; composição de operadores por materialização e pipelining; algoritmos paralelos

2.2.Otimização: medidas de custo; estimativas associadas a planos de avaliação

3.Concorrência e processamento de transações

3.1.Introdução ao controlo de concorrência em bases de dados

3.2.Processamento de transações ACID e níveis de isolamento

3.3.Protocolos baseados em locks e em time-stamps

3.4.Protocolos multiversão

3.5.Esquemas de recuperação em bases de dados

4.Fundamentos de Arquiteturas de Sistemas de Bases de Dados

4.1.Bases de dados centralizadas e cliente-servidor

4.2.Introdução às bases de dados distribuídas: homogéneas e heterogéneas; processamento de perguntas em bases de dados homogéneas; transações em bases de dados distribuídas

3.3.5. Syllabus:

1.DBMSs implementation

1.1.Storage and file structure

1.2.Indexing

1.3.Static and Dynamic Hashing.

2.SQL query processing and optimization

2.1.Processing: general schema; algorithms for relational algebra operators; operators composition – materialisation and pipelining; parallel algorithms

2.2.Optimization: cost measures; estimation of evaluation plans

3.Concurrency and transaction processing

3.1.Introduction to concurrency control in databases

3.2.Transaction processing and isolation levels

3.3.Lock and time-stamp based protocols

3.4.Multiversion protocols

3.5.Recovery in databases

4.Fundamentals of DBMS architecture

4.1.Centralised and client-server databases

4.2.Introduction to distributed databases: homogeneous and heterogeneous; query processing in distributed databases; transactions in distributed databases

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos programáticos seguem muito de perto os objetivos de conhecimento estabelecidos para a unidade curricular, sendo assim clara a coerência entre ambos.

As aptidões objetivo da unidade curricular decorrem da execução do programa em atividades práticas laboratoriais. Assim: a parte do programa sobre implementação de SGBDs é coerente com o objetivo de otimização de bases de dados; a parte sobre processamento e otimização de perguntas é coerente com o objetivo de aptidão para otimização de execução de perguntas; a partes de processamento de transações e de introdução a bases de dados distribuídas contribuem diretamente para as 3ª e 4ª aptidões acima mencionada; o uso avançado de SQL decorre da atividade prática laboratorial ao longo de toda a unidade curricular. As competências são também fomentadas pela forma de avaliação da unidade curricular.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus follows closely the knowledge objectives of the curricular unit, thus making into very clear the coherence among them.

The skills follow from the experimentation in the lab of the subjects in the syllabus. In particular: the experimentation related to the implementation of DBMSs is coherent with the objective of optimization of databases; the experimentation on query processing and optimisation is coherent with the goal of query execution optimisation; the parts on transaction processing and distributed databases contribute directly for the 3rd and 4th skills listed above; the advanced use of SQL follow from the practical activities in the labs during the whole semester.

Competences are also a consequence of the evaluation method of the course.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas e práticas. Nas primeiras são lecionados os conceitos e técnicas relevantes.

Nas aulas práticas são resolvidos problemas e feitas experiências em laboratório de computadores sobre sistemas de gestão de bases de dados, usando o Oracle.

A avaliação de conhecimentos inclui:

- *3 testes individuais teóricos, onde é avaliado o conhecimento que os alunos adquiriram dos conceitos e características das técnicas lecionadas.*
- *1 trabalho prático, elaborado em grupo de 3 estudantes que incide sobre o estudo de sistemas de gestão de bases de dados que não o que é utilizado nas aulas práticas (i.e. que não o Oracle).*

A avaliação do trabalho prático inclui apresentação oral do mesmo, e avaliação pelos pares, de forma anónima.

3.3.7. Teaching methodologies (including evaluation):

The syllabus is taught in theoretical and laboratory classes. In the former, the main concepts and techniques are addressed.

The laboratory classes are dedicated to solving problems, and experimenting the various concepts of the syllabus using the DBMS Oracle.

The evaluation of students include:

- *3 midterm tests where the knowledge acquired by the students on the concepts and characteristics and techniques are assessed.*
- *1 practical projects, made in groups of 3 students, with a study of DBMSs other than the one used in the labs (i.e. other than Oracle).*

The project evaluation includes an oral presentation of it, and anonymous peer-review evaluation.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular identificam a) a aquisição de conhecimento sobre o funcionamento de SGBDs; b) a aquisição de aptidões de melhor uso destes sistemas; e c) algumas competências transversais. O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição dos vários conceitos exposto no programa. Estas matérias são avaliadas em testes individuais, onde os alunos têm um tempo limitado para mostrar o seu grau de assimilação das matérias lecionadas.

Quanto a b) este decorre não só dos conhecimento adquiridos como da experimentação em laboratório e do uso no contexto de um SGBD diferente, no projeto.

As competências em c) são fomentadas pelo projeto e pela sua avaliação. No projeto o estudante terá de trabalhar em grupo e é chamado a fazer uma avaliação crítica do trabalho dos seus colegas, sendo também avaliado por isso.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The objectives of this course identify a) the acquisition of knowledge on the implementation and operation of DBMSs; b) the acquisition of skill for a better usage of DBMSs; and c) some transferable skills.

Objective a) is basically achieved in the theoretical classes, where the main concepts and techniques in the syllabus are explained. These topics are assessed in individual midterm tests, where the students have a limited time to show their level of assimilation of the relevant knowledge.

Regarding b) it follows from the acquired knowledge and also from the experimentation in the lab, and the usage of different DBMSs, as required by the project

The competences in c) are promoted by the project and its evaluation. In the project, the student has to work in a group, and he is required to make a critical evaluation of the work of colleagues from other works, being also evaluated for that.

3.3.9. Bibliografia principal:

Livros de Texto / Text Books

- *Database System Concepts. A. Silberschatz, H. Korth and S. Sudarshan, McGraw-Hill 2010.*
 - *Database Systems: the complete book. H. Garcia-Molina, J Ulmann and J. Widom. Prentice Hall, 2009*
- Manuais /Manuals*
- *Oracle 11g Documentation*

3.3.1. Unidade curricular:

Sistemas Distribuídos / Distributed Systems

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Augusto Legatheaux Martins (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Nuno Manuel Ribeiro Preguiça (Regente) (T-28h; P-112h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Esta é uma UC de consolidação, opcional, em sistemas distribuídos. Esta UC introduz os modelos, métodos e técnicas básicas para o desenvolvimento de sistemas distribuídos. Como pré-requisito, os estudantes devem possuir conhecimentos de algoritmos, programação e redes de computadores.

Saber

- * Problemas e desafios para o desenho de sistemas distribuídos;*
- * Modelos arquiteturais para a construção de sistemas distribuídos;*
- * Modelos e sistemas de comunicação direta e indireta;*
- * Alternativas para a nomeação, incluindo serviços de nomes e diretório;*
- * Soluções fundamentais de ordenação e registo de causalidade de eventos;*
- * Modelos de consistência básicos e soluções de caching e replicação.*

Saber Fazer

- * Desenhar um sistema distribuídos para um problema não trivial;*
- * Utilização de subsistemas de comunicação standard para a programação de sistemas distribuídos;*
- * Implementação de sistemas distribuídos não triviais.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This is an optional consolidation course on distributed systems. It provides the basic knowledge on the models, methods and techniques for developing distributed systems. As prerequisites students should have previous acquaintance with algorithms, programming, and computer networks.

Knowledge

- * Problems and challenges for the design of distributed systems;*
- * Architectural models for building distributed systems;*
- * Models and systems for direct and indirect communication;*
- * Alternatives for naming, including name and directory services;*
- * Essential solutions for ordering and tracking causality of events;*
- * Basic consistency models and solutions for caching and replication.*

Application

- * Design a distributed system for solving a non-trivial problem.*
- * Leverage standard communication sub-systems for programming distributed systems;*
- * Program non-trivial distributed systems.*

3.3.5. Conteúdos programáticos:

1. Introdução

1.1 exemplo; características; desafios

2. Arquiteturas e modelos

2.1 Arquiteturas: cliente/servidor; p2p; proxy

2.2 Modelos de faltas, interação, segurança

3. Sistemas de comunicação direta

3.1 Comunicação ponto-a-ponto

3.2 Multicast

4. Invocação remota

4.1 Modelo

4.2 Interfaces e representação de dados

4.3 Protocolos e semântica na presença de falhas

4.4 Ligação (binding) e concorrência no servidor

5. Invocação remota na Internet

5.1 Web-services

5.2 REST

5.3 Invocação assíncrona (e.g. AJAX) e modelos “push”

6. Sistemas de comunicação indireta

6.1 Comunicação em grupo

6.2 Sistemas produtor/subscritor

6.3 Sistemas de filas de mensagens

7. Sistemas de Nomes

7.1 Problemática e conceitos

7.2 Serviços de nomes

7.3 Serviços de diretório

8. Tempo

8.1 Relógios físicos

8.2 Relógios lógicos

8.3 Relógios vectoriais

8.4 Vetores versão

9. Introdução à replicação e consistência

9.1 Caching

9.2 Replicação primário/secundário

3.3.5. Syllabus:

1. Introduction

1.1 Examples, characteristics, challenges

2. Architectures and models

2.1 Architectures: client/server, p2p, proxy

2.2 Fault, interaction and security models

3. Direct communication systems

3.1 Point-to-point communication

3.2 Multicast

4. Remote invocation

4.1 Model

4.2 Interfaces and data representation

4.3 Protocols and semantics in the presence of faults

4.4 Binding and concurrency in the server

5. Remote invocation in the Internet

5.1 Web-services

5.2 REST

5.3 Asynchronous invocation (e.g. AJAX) and push models

6. Indirect communication systems

6.1 Group communication

6.2 Publish/subscribe

6.3 Message queues

7. Naming in distributed systems

7.1 Problems and concepts

7.2 Name services

7.3 Directory services

- 8. Time
- 8.1 Physical clocks
- 8.2 Logical clocks
- 8.3 Vector clocks
- 8.4 Version vectors

- 9. Introduction to replication and consistency
- 9.1 Caching
- 9.2 Primary/backup replication

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A UC visa fornecer um conjunto de conhecimentos e competências que lhe permitem desenvolver sistemas distribuídos com alguma complexidade. Para atingir esse objetivo, a UC começa por apresentar os principais problemas e desafios que se colocam à construção dum sistema distribuído, estudando de seguida os modelos e arquiteturas que podem ser usadas na construção dum sistema distribuído.

De seguida, estudam-se de forma sistemática as técnicas de comunicação base usadas na criação de sistemas distribuídos.

O programa da unidade curricular prossegue abordando tópicos fundamentais usados na construção de sistemas distribuídos: sistemas de nomes, tempo e técnicas de replicação.

Na componente prática, os conceitos estudados na parte teórica serão colocados em prática desenvolvendo sistemas distribuídos de complexidade crescente, permitindo aos estudantes ganhar experiência no desenvolvimento deste tipo de sistemas.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course aims to provide students with a set of knowledge and know-how that allows them to develop distributed systems with some complexity. To teach this goal, the course starts by presenting the main problems and challenges of distributed systems, and the architectures and models used in the design of distributed systems.

The course continues by systematically studying communication solutions used for developing distributed systems and other fundamental topics: naming in distributed systems, time and replication techniques.

In the labs, the topics studied in the lectures will be exercised in the development of distributed systems with increasing complexity, allowing students to gain experience in programming distributed systems.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas expõe-se e discute-se o programa da cadeira, apresentando os problemas e discutindo as respetivas soluções recorrendo a exemplos concretos da utilização de sistemas distribuídos.

As aulas práticas acompanham os dois projetos da cadeira, fornecendo aos alunos experiência no desenvolvimento de sistemas distribuídos usando as técnicas estudadas. Os projetos são constituídos por uma sequência de passos que seguem a matéria abordada. Nas aulas práticas apresentam-se exemplos e inicia-se a resolução dos vários passos dos projetos, e presta-se apoio à sua realização.

A avaliação será baseada nos seguintes componentes: dois testes (35% cada); dois projetos de programação, resolvidos em grupo (15% cada).

3.3.7. Teaching methodologies (including evaluation):

In lectures, the key points of the syllabus are presented, using examples of existing distributed systems whenever possible. This presentation should be complemented with studying the recommended book.

The laboratory classes target the presentation and resolution of two programming projects, providing students with experience in using the studied techniques for programming distributed systems. Projects will include a sequence of steps that follow the topics studied. Examples will be presented in laboratory classes, where each step of the project will be started.

The course grade will be computed based on the following components: two quizzes (35% each) and two programming projects solved in team (15% each).

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As metodologias de ensino têm como objetivo permitir conhecer os modelos e técnicas usadas no desenvolvimento de sistemas distribuídos, assim como permitir aos estudantes colocar esta mesmas técnicas em prática. Tal é conseguido através da exposição dos conceitos e técnicas fundamentais nas aulas teóricas e sua consolidação e colocação em prática nas aulas práticas, através da apresentação de exemplos e realização de dois projetos de programação.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The teaching methodologies have the goal of allowing students to know the models and techniques used in the developments of distributed systems, as well as put this knowledge in practice in the development of real systems. This is achieved by presenting the fundamental concepts and techniques in lectures. In labs, the knowledge obtained in lectures in consolidate and exercised by the presentation of examples and the execution of two programming projects.

3.3.9. Bibliografia principal:

*Distributed Systems: Concepts and Design
George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair
Publisher: Addison Wesley; 5th edition
ISBN-13: 978-0132143011*

Mapa IV - Algoritmos e Sistemas Distribuídos / Distributed Algorithms and Systems

3.3.1. Unidade curricular:

Algoritmos e Sistemas Distribuídos / Distributed Algorithms and Systems

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Rodrigo Seromenho Miragaia Rodrigues (T-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Sérgio Marco Duarte (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

A unidade curricular (UC) visa consolidar e desenvolver os conhecimentos na área de Sistemas Distribuídos para a compreensão e conceção de sistemas descentralizados complexos. Procura-se um bom domínio das ideias fundamentais que estão na base de sistemas atuais, de uso corrente, e das técnicas que potencialmente serão importantes para os sistemas a desenvolver no futuro.

A UC tem uma forte componente algorítmica, mas é também acompanhada de projetos de programação que põem em prática os conceitos fundamentais que são nela ensinados.

Saber:

- *Conceitos de base para a análise e síntese de algoritmos distribuídos.*
- *Abstrações fundamentais para a construção de sistemas distribuídos e a sua realização algorítmica.*
- *Técnicas para melhorar a fiabilidade e a escalabilidade dos sistemas distribuídos.*

Saber Fazer:

- *Conceção de algoritmos distribuídos e a sua aplicação no desenvolvimento de sistemas distribuídos.*
- *Análise de algoritmos distribuídos.*
- *Programação de sistemas distribuídos.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The central goal of this course is to consolidate the knowledge of the students in the area of distributed systems, by gaining a better understanding of the algorithms that are used by the systems that are in use today, and by learning the fundamental ideas that were useful in their design, and that might be useful in the design of the distributed systems that will be used in the future. The course has a strong algorithmic component, but is also accompanied by a set of programming projects, which put in practice the fundamental concepts learned in lectures.

Knowledge:

- Basic concepts to the analysis and synthesis of distributed algorithms
- Fundamental abstractions for building distributed systems and the algorithms that are used to realize them
- Techniques for improving the reliability and scalability of distributed systems

Application:

- Designing distributed algorithms and applying them in to build distributed systems
- Analyzing distributed algorithms
- Programming distributed systems

3.3.5. Conteúdos programáticos:

1. Modelos de computação distribuída

1.1. Modelos de processos, falhas, primitivas criptográficas e comunicação.

1.2. Modelo síncrono, assíncrono, e parcialmente síncrono.

2. Replicação e tolerância a faltas, e respetivas abstrações fundamentais

2.1. Especificação dos sistemas de replicação: diferentes modelos de consistência

2.2. Protocolos de replicação

2.3. Estudo das abstrações de suporte à replicação: consenso e variantes

3. Escalabilidade

3.1. Sistemas peer-to-peer

3.2. Outras técnicas para a construção de sistemas escaláveis.

3.3. Estude de casos.

4. Transações distribuídas

4.1. Two-phase commit

4.2. Three-phase commit

4.3. Técnicas de logging para tolerância a faltas

3.3.5. Syllabus:

1. Distributed computation models

1.1. Modeling processes, faults, crypto primitives and communication.

1.2. Synchronous, asynchronous and partially synchronous models.

2. Replication, fault tolerance, and respective fundamental abstractions

2.1. Specifying replicated systems: different consistency levels

2.2. Replication protocols

2.3. Studying the basic abstractions to support replication: consensus and variants

3. Scalability

3.1. Peer-to-peer systems

3.2. Other techniques for scaling distributed systems

3.3. Case studies.

4. Distributed transactions

4.1. Two-phase commit

4.2. Three-phase commit

4.3. Logging for fault tolerance

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O conteúdo programático da cadeira começa por fornecer aos alunos as bases fundacionais para a análise e síntese de algoritmos distribuídos. De seguida, são estudadas abstrações relacionadas com técnicas de tolerância a faltas, e os alunos têm oportunidade de aprender a sua realização em algoritmos distribuídos e analisar a sua correção e desempenho. De seguida, os alunos estudam várias alternativas para a conceção de sistemas distribuídos escaláveis, aprendendo as técnicas que suportam esta característica dos algoritmos usados nestes sistemas. Finalmente, são estudadas as transações distribuídas, onde os alunos são expostos a esta técnica importante na construção de sistemas distribuídos robustos.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The course program starts by giving student the foundations required for analyzing and synthesizing distributed algorithms. Next, the abstractions related to fault tolerance techniques are studied, and the students have the opportunity to learn how they are realized in distributed algorithms and to analyze their correction and performance. Next, several ways to design scalable distributed systems are studied. Finally, the topic of distributed transactions is studied, where students are exposed to this important technique for building robust distributed systems.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas explicam-se e discutem-se os sucessivos tópicos do programa da cadeira. A abordagem aos problemas e à explicação dos conceitos e soluções tenta ser rigorosa, mas evitando um formalismo excessivo. Nos vários tópicos serão estudados exemplos de sistemas que integram soluções para as problemáticas estudadas.

As aulas práticas acompanham os projetos da cadeira, fornecendo aos alunos a experiência sobre como é que os conceitos que estão a ser aprendidos na cadeira podem ser aplicados na prática. Desta forma, nas aulas práticas faz-se uma introdução à fase atual do projeto, dão-se os primeiros passos na sua resolução, e presta-se apoio à sua realização.

Componentes da avaliação

- *Dois testes.*
- *Dois projetos de programação.*
- *Vários checkpoints de cada projeto que permitem dar um incentivo ao trabalho contínuo durante a cadeira e distribuir a carga de estudo e trabalho de forma mais uniforme ao longo do semestre.*

3.3.7. Teaching methodologies (including evaluation):

In lectures the various topics are explained and discussed. The approach to this will be rigorous, while avoiding an excessive formalism. A set of selected systems will be studied for illustrating the techniques discussed in the different topics.

Laboratory classes follow the course projects closely, giving students the experience of how the concepts learned in lecture can be applied in practice. Thus, each lab class will give an introduction to the current stage of the project, provide some initial steps in how to solve it, and provide support to students in more advanced stages of its resolution.

Final grade calculation

- *Two quizzes (midterm and term)*
- *Two programming projects*
- *Several checkpoints in each project to given an incentive to work continuously throughout the semester, and to distribute the workload uniformly.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As metodologias de ensino têm como objetivo permitir a compreensão aprofundada sobre a análise e síntese de algoritmos distribuídos, com ênfase particular nos sistemas replicados, e também compreender as técnicas para melhorar a escalabilidade dos sistemas distribuídos. Tal será conseguido através da uma integração cuidada entre os conhecimentos explicados na aula teórica e a aprendizagem dos conceitos através da programação, no contexto das aulas práticas e do projeto, de sistemas que fazem uso dos conceitos aprendidos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The teaching methodologies aim at providing an in-depth understanding concerning the analysis and synthesis of distributed algorithms with a particular emphasis on replicated systems, as well as promoting an understanding of the techniques for improving the scalability of distributed systems. The above goals will be achieved through a careful integration of the themes presented and discussed in the lectures and the programming assignments performed by the students in the context of the lab sessions and projects, where the theoretical concepts and approaches are exercised in practice.

3.3.9. Bibliografia principal:

Bibliografia de base / Course reading

- *C. Cachin, R. Guerraoui, L. Rodrigues "Introduction to Reliable and Secure Distributed Programming", 2nd edition, Springer, 2011.*
- *Conjunto de artigos selecionado.*

Bibliografia complementar / Complementary reading

- *H. Attiya and J. Welch. Distributed Computing: Fundamentals, Simulations, and Advanced Topics (2nd Ed.) . Wiley 2004.*
- *S. Mullender (editor) Distributed Systems, Second Edition, ACM Press, Addison-Wesley, MA, 1994.*
- *N. Lynch. Distributed Algorithms Morgan Kauffman, 1996.*
- *A.S. Tanenbaum and M. van Steen. Distributed Systems. Principles and Paradigms. (2nd Ed.) Prentice Hall, 2007.*
- *Rodrigo Rodrigues, Peter Druschel. Peer-to-Peer Systems. Communications of the ACM. Vol. 53 No. 10, Pages 72-82.*

Mapa IV - Arquitetura e Implementação de Sistemas de Operação / Operating Systems Design and Implementation

3.3.1. Unidade curricular:

Arquitetura e Implementação de Sistemas de Operação / Operating Systems Design and Implementation

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Alberto Cardoso e Cunha (apenas Responsável não tem horas de contato)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Manuel dos Santos Lourenço (T-28h; P-28h) (Regente)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

A disciplina visa desenvolver nos alunos um conhecimento aprofundado sobre os Sistemas de Operação (SO), nos seus aspetos de conceção e organização interna, bem como nas técnicas e algoritmos utilizados na sua implementação.

Saber:

- *Conceção e organização interna, técnicas e algoritmos para a implementação de SO, nas seguintes dimensões:*
- *gestão de recursos (processador, memória e periféricos)*
- *sistemas de ficheiros*
- *processos e threads, mecanismos de proteção.*
- *suporte da virtualização, multiprocessamento e de sistemas específicos embutidos e de tempo real.*

Saber Fazer:

- *identificar os compromissos de realização do núcleo e serviços do SO e estimar os seus comportamentos, face a requisitos específicos*
- *implementar módulos internos do SO e lidar com a sua complexidade*
- *avaliar experimentalmente diferentes sistemas dependendo das soluções arquitetónicas e algoritmos utilizados.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This course aims at providing the students with an in-depth knowledge on the Operating System (OS) internal organization, and the techniques and algorithms used in its implementation.

Knowledge:

- *Issues in the design and organization of OS, and the techniques and algorithms for implementing OS, including:*
- *Resource management (processors, memory and input/output devices).*
- *File systems.*
- *Processes and threads. Protection mechanisms.*
- *OS support for virtualization, multiprocessing and embedded systems*

Application:

- *Identify the trade-offs in the design and implementation of OS kernel and services, and how to estimate their expected behaviour under specific requirements*
- *ability to implement/modify internal OS modules and handle their complexity*
- *ability to experimentally evaluate distinct systems, depending on their internal organizations and strategies*

3.3.5. Conteúdos programáticos:

1. Introdução

- *Funções e organização interna do núcleo e serviços.*

2. Gestão de processos

- *Estratégias e mecanismos de gestão de processos/threads.*
- *Interação entre processos/threads. Partilha de recursos*
- *Escalonamento com multiprogramação e multiprocessamento. Requisitos de tempo real*

3. Gestão de memória

- *Hierarquia de memórias. Memória virtual: swapping, paginação e segmentação.*
- *Estratégias e algoritmos. Paginações a pedido, working sets e trashing, proteção e partilha*
- *Mapeamento de ficheiros em memória.*

4. Sistemas de ficheiros e entradas/saídas

- *Gestão de meta-dados, de dados e do espaço livre/ocupado.*

- *Disponibilidade, verificação e recuperação.*
- *Suportes físicos alternativos.*
- *Entradas e saídas: estratégias e mecanismos. Device drivers.*
- *Análise de sistemas reais*

5. Virtualização, multiprocessamento e SO dedicados

- *Suporte hw/sw à virtualização.*
- *Multiprocessadores. Single system image.*
- *SO embutidos: restrições de memória, tempo e energia.*

3.3.5. Syllabus:

1. Introduction

- *Functions and internal organization of the OS kernel and services*

2. Process Management

- *Strategies and mechanisms for process/thread management and implementation*
- *Process/thread interactions and resource sharing.*
- *Scheduling under multiprogramming, multiprocessing, and real-time environments.*

3. Memory Management

- *Memory hierarchy, Virtual memory: Swapping, paging and segmentation.*
- *Strategies and algorithms. On-demand paging, working sets and thrashing, protection and sharing.*
- *Memory-mapped files*

4. File Systems and Input/output Systems

- *Meta-data and data management. Disk space management.*
- *Availability, verification and recovery.*
- *Alternative storage media.*
- *Input/output strategies and mechanisms. Device drivers.*
- *File system case studies.*

5. Virtualization, multiprocessing, and dedicated embedded systems

- *Hw and Sw support for virtualization.*
- *Multiprocessor OS. Single System Image.*
- *OS for embedded systems: memory, time, and power constraints.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos programáticos cobrem os principais tópicos referentes à implementação de SO, na perspectiva algorítmica e tecnológica. As aulas teóricas motivam para o estudo de cada tópico principal e introduzem os principais problemas a endereçar e as técnicas e algoritmos utilizados para os abordar. O trabalho experimental, baseado numa mistura de laboratórios fechados e abertos, incluem a realização de projetos de tamanho médio que permitirão consolidar os conhecimentos adquiridos durante as aulas teóricas e por autoestudo, e cobrem os principais tópicos referentes ao desenho e implementação de SO e à sua avaliação experimental: gestão de processos, gestão de memória, sistemas de ficheiros e gestão de E/S.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The course syllabus covers the main topics on OS implementation, both for algorithmic and technological perspectives. The lectures motivate for the study of each main topic and introduce the main issues that have to be addressed and techniques and algorithms to overcome them. The experimental work, based in a mix of closed and open laboratories, includes mid-size project assignments that shall consolidate the knowledge acquired during lectures and self-study and covers the main topics in OS design and implementation and their experimental evaluation: process management, memory management, file systems and I/O management.

3.3.7. Metodologias de ensino (avaliação incluída):

A metodologia de ensino da disciplina inclui aulas teóricas e elaboração de projetos com acompanhamento parcial em aulas práticas laboratoriais. As aulas teóricas motivam para o estudo dos componentes arquiteturais do SO, identificam os seus objetivos e problemas, e apresentam algoritmos e técnicas utilizadas para os endereçar.

As aulas práticas proporcionam tempo de contato privilegiado com os docentes da disciplina, informam sobre os detalhes dos projetos a realizar, apoiam os alunos na sua adaptação ao ambiente HW e SW utilizado, disponibilizando também algum tempo para realização dos projetos

Componentes de avaliação:

1. Testes individuais ou exame final (50 %).
2. Projetos de programação a desenvolver em grupo, incluindo relatórios escrito (50 %).

3.3.7. Teaching methodologies (including evaluation):

The course teaching methodology includes lectures and project assignments, partially developed in the laboratory practical classes. The lectures motivate for the study of the OS architectural components, identify their objectives and difficulties, and present the techniques and algorithms used to address them.

The practical classes provide privileged contact time with de TAs, inform about the details of the project assignments, support the student in adapting to the experimental HW and SW environment used, and also provide some time for project development.

Assessment Components:

1. Individual tests or final exam (50 %).
2. Group project assignments, including written reports (50 %).

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A separação entre aulas teóricas, práticas e projetos de programação permitem que o aluno obtenha uma compreensão profunda dos objetivos dos SO, das suas soluções arquiteturais, e das técnicas e algoritmos utilizados para implementar essas soluções. Os laboratórios fechados/assistidos providenciam horas adicionais de contacto com membros da equipa docente e com outros colegas, e permita a troca e partilha de experiências. Os projetos de programação permitem ao estudante consolidar os conhecimentos adquiridos nas aulas e por autoestudo, através da implementação e experimentação de técnicas e algoritmos utilizados em SO reais e educacionais.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The separation between lectures, closed labs and project assignments allow the student with achieve a complete understanding of the objectives of OS and their architectural solutions, and of the algorithms and techniques used when implementing those solutions. The closed labs provide the students with additional contact hours with teaching team members and other colleagues, and allow the exchange and sharing of experiences. The project assignments allow the student to consolidate the knowledge acquired in the lectures and self-study, by implementing and experimenting techniques and algorithms in real-world or educations OS.

3.3.9. Bibliografia principal:

Main references:

- Thomas W. Doeppner. *Operating Systems In Depth: Design and Programming*. Wiley & Sons, 2010.
- Peter Baer Galvin and Greg Gagne. *Operating System Concepts, 8th Edition*. Avi Silberschatz, John Wiley & Sons, 2010.

Mapa IV - Codificação de Informação / Information Coding

3.3.1. Unidade curricular:

Codificação de Informação / Information Coding

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Abílio Duarte de Medeiros (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Carmen Pires Morgado (Regente) (T-28h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

O objetivo da unidade curricular é apresentar os princípios e métodos usados na codificação da informação para a sua transmissão e armazenamento. São estudados os conceitos provenientes da Teoria de Informação e a forma são utilizados para fundamentar as técnicas usadas na compressão de dados, criptografia e deteção e correção de erros.

Saber:

- *Fundamentos da Teoria da Informação e as suas contribuições para a codificação da informação;*
 - *Principais técnicas de compressão de dados e de deteção e correção de erros;*
 - *Algoritmos usados na criptografia simétrica e de chave pública e sua análise à luz da Teoria da Informação.*
- Saber Fazer:**
- *Analisar e determinar a eficiência das diferentes técnicas de compressão de acordo com o tipo de dados manipulados;*
 - *Analisar as implementações de algoritmos usados em criptografia*
 - *Desenvolvimento de versões simplificadas de algoritmos de compressão de dados, deteção e correção de erros e criptografia.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The main objective of the course is to present the principles and methods related to information storage and transmission; the fundamentals of information theory are used to understand the principles of data compression, error detection and correction and cryptography.

Knowledge:

- *Fundamental results of Information Theory and its contributions to information coding;*
- *Main techniques for data compression, error detection and correction;*
- *Algorithms used for symmetric and public-key encryption and its analysis according to the Information Theory concepts.*

Application:

- *Analyze and determine the efficiency of different compression techniques according to the type of data handled;*
- *Analyze different techniques of cryptography and its different uses;*
- *Development of simplified versions of algorithms for data compression, error correction and detection and cryptography.*

3.3.5. Conteúdos programáticos:

1. Introdução à codificação de informação

- *Blocos constituintes de um sistema digital de comunicação/armazenamento da informação*
- *Definição de informação. Entropia. Teorema de Shannon da codificação da fonte*
- *Codificação de dados: códigos de descodificação única, instantânea e de condição prefixa*

2. Compressão de informação

- *Codificação estatística e aritmética*
- *Codificação baseada em dicionários*
- *Codificação baseada no contexto*
- *Análise de alguns programas de compressão*
- *Codificação de imagem, som e vídeo*

3. Algoritmos criptográficos

- *Relação da criptografia com a teoria da informação. Noção de segredo perfeito*
- *Cifras simétricas*
- *Cifras de chave pública*
- *Assinaturas digitais*
- *Esteganografia*

4. Deteção e correção de erros

- *Canais com ruído. Capacidade de um canal*
- *Métodos de deteção e correção de erros: códigos lineares, cíclicos, BCH e de convolução*
- *Aplicações de métodos de deteção e correção de erros: ISBN e similares, RAID, formatos CD/DVD*

3.3.5. Syllabus:

1. Fundamentals of information coding

- *Building blocks of a data communication/storage system*
- *Entropy of an information source. Shannon theorem of source codification*
- *Coding theory: unique, instantaneous decoding, prefix-free codes*

2. Information compressions

- *Statistical and Arithmetic coding*
- *Dictionary coding*
- *Contextual coding*
- *Analysis of daily-used compression programs*
- *Image, video and sound compression*

3. Cryptography algorithms

- *Information theory and cryptography*
- *Symmetric ciphers*

- *Public-key algorithms*
- *Steganography*
- 4. *Error detection and correction*
 - *Noisy channels; Channel capacity theorem*
 - *Error detection and correction methods: linear, cyclic, BCH and convolution codes*
 - *Applications of error detection and correction: ISBN and related codes, RAID, CD/DVD formats*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O conteúdo programático desta unidade curricular apresenta aos alunos os conceitos fundamentais da Teoria da Informação dando uma visão dos diferentes componentes envolvidos num sistema de transmissão e armazenamento da informação de informação. São apresentados diversos algoritmos e técnicas de compressão e correção de dados e dos seus fundamentos teóricos. Diversas técnicas de criptografia e da sua utilização são analisadas com especial ênfase na criptografia de chave pública.

O programa desta unidade curricular apresenta também aos alunos diferentes técnicas para a utilização, o mais eficaz possível, dos sistemas de comunicação e armazenamento de informação (codificação/compressão de informação). Sendo analisadas algumas técnicas para a transmissão com fiabilidade e segurança de dados num canal de comunicação (códigos para deteção e correção de erros e criptografia).

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course introduces students to the fundamental concepts of Information Theory by giving an overview of the different components involved in a system of information transmission. Several algorithms and compression techniques and correction of data are presented as well as its theoretical foundations. Several encryption techniques and their use are discussed with special emphasis on public key cryptography and private. The program of this course also presents students to different techniques to use in a most effective way, communication systems and storage (coding / data compression). Some techniques for transmission with reliability and safety data in a communication channel (codes for detection and error correction and encryption), are discussed.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas explicar-se-ão e discutir-se-ão os sucessivos tópicos do programa da unidade curricular. Sendo a apresentação dos tópicos apresentados, reforçada através de diversos exemplos práticos.

Nas aulas práticas serão desenvolvidos diversos trabalhos práticos de programação, cujo objetivo é aplicar e consolidar os conceitos que foram aprendidos nas aulas teóricas. Em particular recorrendo à implementação de versões simplificadas dos algoritmos e técnicas apresentadas nas aulas teóricas.

Componentes da avaliação:

- *Dois testes que versam os conhecimentos quer teóricos quer práticos dos conteúdos programáticos apresentados nesta unidade curricular;*
- *Dois projetos de programação de média dimensão correspondendo à implementação de alguns dos algoritmos apresentados. O primeiro está relacionado com a compressão de dados e no segundo em que se implementa um método criptográfico.*

3.3.7. Teaching methodologies (including evaluation):

At the theoretical classes the successive threads of the program of the course, will be explained and discussed. The presentation of the topics presented will be reinforced through several practical examples.

At the laboratory classes many practical programming tasks will be developed, this aim to implement and consolidate the concepts that were learned in the lectures, in particular using the implementation of simplified versions of algorithms and techniques presented.

Assessment components

- *Two mid-term tests encompassing theory and methods of the curricular unit.*
- *Two medium-sized programming projects where students will implement some of the algorithms presented (the first one will focus upon data compression techniques and the second one on encryption techniques).*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As metodologias de ensino são tradicionais e envolvem: exposição dos aspetos essenciais da matéria, resolução de exercícios com modelização e análise dos algoritmos e técnicas de compressão de dados, correção de erros e criptografia. A realização de trabalhos práticos de laboratório e os dois projetos envolvem a implementação de pequenos programas que ilustram de forma simplificada a maioria dos algoritmos e técnicas apresentadas nas aulas.

Cada aula de laboratório será sempre antecedida de aula teórica onde foram expostos os temas que constam

dos trabalhos práticos.

Os trabalhos a realizar em laboratório têm como principal objetivo contribuir para a o esclarecimento de dúvidas e para uma consolidação dos conhecimentos apresentados nas aulas teóricas. Constituindo esses trabalhos uma ferramenta de estudo essencial para a aprendizagem do aluno.

Os projetos a desenvolver pelos alunos envolvem para além de uma componente de implementação uma exposição, sob a forma de relatório escrito e sob a forma de discussão oral, que permite expor as dificuldades habitualmente sentidas pelos alunos, na apresentação escrita e oral.

Nos testes pretende-se que o aluno tenha a capacidade de identificar e relacionar as diferentes técnicas apresentadas, tendo a capacidade de aplicar os conceitos aprendidos a situações concretas de utilização.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The teaching methods are traditional and involve: exposure of the essential aspects of the matter; problem solving with modeling and analysis of algorithms; and techniques for data compression, error correction and encryption. The practical work of the laboratory and the two projects involve the implementation of small programs that illustrate, in a simplified form, algorithms and techniques presented at classes.

Each laboratory class will always be preceded by lecture in which the topics of the practical work were exposed.

The work to be performed in laboratories has as main objective to contribute to the clarification of doubts and a consolidation of knowledge presented in lectures. These tasks constitute an essential study tool for student learning.

The projects to be developed by students will involve in addition to a component implementation exposure in the form of a written report and an oral discussion, which allows you to expose the difficulties commonly experienced by students in written and oral presentation.

With the tests, we intend that students will have the ability to identify and relate the different techniques presented, with the ability to apply the studied concepts to concrete situations of use.

3.3.9. Bibliografia principal:

Referências principais / Course reading:

- D. Salomon, Coding for Data and Computer Communications, Springer, 2005

- R.Wells, Applied Coding and Information Theory for Engineers, Prentice-Hall, 1998

Referências complementares / Complementary reading:

- A. Brunen, M. Forcinito, Cryptography, Information Theory. and Error-correction, Wiley, 2005

- T. Cover, J. Thomas, Elements of Information Theory 2nd Ed, Wiley, 2006

Mapa IV - Programação com Restrições / Constraint Programming

3.3.1. Unidade curricular:

Programação com Restrições / Constraint Programming

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Manuel Corrêa Calvente de Barahona (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Francisco de Moura e Castro Ascensão de Azevedo (Regente) (T-14h; P-14h)

Jorge Carlos Ferreira Rodrigues da Cruz (T-14h; P-14h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

• Problemas de satisfação de restrições em domínios contínuos e conjuntos.

• Tipos de restrições em domínios contínuos e conjuntos.

• Propagação de restrições e tipos de consistência.

• Algoritmos para satisfação de restrições em domínios contínuos e conjuntos.

• Linguagens atuais e suas limitações.

Saber Fazer:

• Modelação de problemas de satisfação de restrições em domínios contínuos e conjuntos.

• Desenvolvimento de algoritmos de propagação de restrições em domínios contínuos.

- *Programação com o Cardinal.*
 - *Aplicação de inferências e lógicas multivalor.*
 - *Uso de pesquisa local com conjuntos de dependências.*
- Competências Complementares:**
- *Compreensão de semânticas operacionais.*
 - *Formulação de modelos de restrições.*
 - *Análise e explicação de resultados.*
 - *Capacidade de pesquisa de literatura.*
 - *Capacidade de abstração e generalização.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Constraint satisfaction problems in continuous domains and sets.*
- *Types of constraints in continuous domains and sets.*
- *Constraint propagations and types of consistency.*
- *Algorithms for constraint satisfaction in continuous domains and sets.*
- *Constraint programming languages and their limitations.*

Application:

- *Modelling of constraint satisfaction problems in continuous domains and sets.*
- *Implementation of constraint propagation algorithms for continuous domains.*
- *Constraint programming with Cardinal.*
- *Usage of inference and multivalued logic.*
- *Local search with dependency sets.*

Soft-skills:

- *Understanding of operational semantics.*
- *Constraint modelling.*
- *Analysis and explanation of results.*
- *Specialised literature searching.*
- *Abstraction and generalization capabilities.*

3.3.5. Conteúdos programáticos:

- 1. Introdução aos problemas de restrições em domínios contínuos.*
- 2. Análise de intervalos.*
 - 2.1. Aritmética de intervalos.*
 - 2.2. Método de Newton com intervalos.*
- 3. Propagação de restrições em domínios contínuos.*
 - 3.1. Funções de redução de domínio.*
 - 3.2. Algoritmo de propagação.*
 - 3.3. Consistência.*
- 4. Resolução de problemas em domínios contínuos.*
 - 4.1. Técnicas de modelação.*
 - 4.2. Linguagens e ferramentas.*
 - 4.3. Exemplos.*
- 5. Introdução aos problemas de restrições em conjuntos.*
 - 5.1. Conjuntos e multiconjuntos.*
 - 5.2. Exemplos e aplicações.*
 - 5.3. Resolvedores de restrições.*
- 6. Cardinal.*
 - 6.1. Semântica operacional.*
 - 6.2. Atribuição de valores.*
 - 6.3. Noções básicas de implementação.*
 - 6.4. Exemplos.*
- 7. Aplicações.*
 - 7.1. Circuitos digitais.*
 - 7.2. Lógicas multivalor.*
 - 7.3. Resultados e comparação com SAT.*
- 8. Otimização.*
 - 8.1. Branch & Bound.*
 - 8.2. Extensões da linguagem Cardinal*
 - 8.3. Pesquisa local.*

3.3.5. Syllabus:

1. *Introduction to continuous constraint satisfaction problems.*
2. *Interval analysis.*
 - 2.1. *Interval arithmetic.*
 - 2.2. *Interval Newton method.*
3. *Constraint propagation in continuous domains.*
 - 3.1. *Narrowing functions.*
 - 3.2. *Propagation algorithm.*
 - 3.3. *Consistency.*
4. *Problem solving in continuous domains.*
 - 4.1. *Modeling techniques.*
 - 4.2. *Languages and tools.*
 - 4.3. *Examples.*
5. *Introduction to constraint satisfaction problems over sets.*
 - 5.1. *Sets and multisets.*
 - 5.2. *Examples and applications.*
 - 5.3. *Set constraint solvers.*
6. *Cardinal.*
 - 6.1. *Operational semantics.*
 - 6.2. *Labeling.*
 - 6.3. *Implementation notes.*
 - 6.4. *Examples.*
7. *Applications.*
 - 7.1. *Digital circuits.*
 - 7.2. *Multivalued logics.*
 - 7.3. *Results and comparison with SAT.*
8. *Optimization.*
 - 8.1. *Branch & Bound.*
 - 8.2. *Cardinal extensions.*
 - 8.3. *Local search.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Nesta unidade curricular aprofundam-se conhecimentos referidos como opcionais no ACM 2008 CS Curriculum na área de Sistemas Inteligentes (IS/AdvancedSearch [elective]), e que complementam os introduzidos em unidades anteriores, nomeadamente em "Pesquisa e Otimização".

A resolução de problemas complexos de decisão e de otimização tem sido muito melhorada pelos avanços recentes na área de programação por restrições, não apenas em domínios finitos, mas também em domínios contínuos e estruturados.

A matéria lecionada insere-se claramente nos objetivos de dar a conhecer aos estudantes os avanços recentes mas já consolidados nesta área, que lhes permitam identificar oportunidades de aplicação destas técnicas em variadas situações práticas, modelar os problemas identificados e resolvê-los de forma eficiente.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course addresses in more depth the topics of search referred to as elective in the ACM 2008 CS Curriculum in the area of Intelligent Systems (IS/AdvancedSearch [elective]), and that complement those introduced in previous courses, namely "Pesquisa e Otimização".

Solving complex decision and optimisation problems have undergone significant improvements in recent years, taking advantage of corresponding advances in constraint programming, not only in finite domains, but also in continuous and structured domains.

The syllabus clearly fulfil the objectives of the course, which is meant to provide the students with the advanced, if consolidated, knowledge in this area, allowing them to detect opportunities to apply it in many situations, modelling the identified problems, and efficiently solve them.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas e práticas. Nas primeiras são lecionados os conceitos e técnicas relevantes bem como a forma como estão implementados na linguagem e nos sistemas de modelação e resolução utilizados.

Nas aulas práticas são resolvidos problemas retirados de benchmarks típicos destas áreas, sendo explorados e desenvolvidos modelos e técnicas de pesquisa apropriados.

A avaliação de conhecimentos inclui:

- *2 testes individuais teóricos, onde é avaliado o conhecimento que os alunos adquiriram dos conceitos e características das técnicas lecionadas.*
- *2 trabalhos práticos que incidem sobre a resolução de problemas de restrições e otimização não triviais, e*

avaliam a capacidade dos alunos em utilizar e adaptar algoritmos para restrições em domínios contínuos e estruturados, explorados nas aulas práticas em problemas mais simples.

3.3.7. Teaching methodologies (including evaluation):

The syllabus is taught in theoretical and laboratory classes. In the former, the main concepts and techniques are addressed, together with their implementation in the modelling and execution languages and systems adopted.

The laboratory classes are dedicated to solving problems, selected from typical benchmarks in these areas, using appropriate models and search algorithm.

The evaluation of students include:

- 2 mid/final term tests, where the knowledge acquired by the students on the concepts and characteristics of the search techniques are assessed.*
- 2 practical projects, that focus on the resolution of non-trivial constraint and optimisation problems, and assess the ability of the students to use and adapt algorithms for constraints over continuous and structured domains, explored in the lab classes on simpler problems.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular identificam a) a aquisição de conhecimento recentemente desenvolvido para a satisfação de problemas de decisão e de otimização sujeitos a restrições em domínios contínuos e estruturados; b) a experimentação da resolução de problemas, com base em modelos e algoritmos apropriados; e c) avaliação da adequação e eficiência das soluções alternativas, identificando-se as mais apropriadas para diferentes situações.

O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição dos várias conceitos e técnicas de programação de restrições nos domínios considerados, sendo estudadas as suas principais propriedades (por exemplo, complexidade). Estes conhecimentos são avaliados em testes individuais, onde os alunos têm um tempo limitado para mostrar o seu grau de assimilação das matérias lecionadas.

O comportamento de diferentes modelos e técnicas de pesquisa a aplicações concretas, requer uma aprendizagem mais baseada na experimentação, já que resultados teóricos existentes (por exemplo, o estudo da complexidade no pior caso) são pouco informativos. Para esse efeito, essa experimentação é feita nas aulas práticas em problemas razoavelmente simples, sendo avaliada através de trabalhos em aplicações não triviais, e que obrigam os estudantes a desenvolver capacidades de análise e de decisão bastante importantes nesta área.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The objectives of this course identify a) the acquisition of recently developed knowledge to solve constraint-based decision and optimisation in continuous and structured domains; b) the experience obtained in solving such problems, adopting appropriate algorithms; and c) assessment of the adequacy and efficiency of alternative approaches, with the identification of the most appropriate for different situations.

Objective a) is basically achieved in the theoretical classes, where the main concepts and techniques from both constraint programming and constrained based local search are explained, together with the study of their main characteristics (e.g. complexity). These topics are assessed in 2 individual mid/final-term tests, where the students have a limited time to show their level of assimilation of the relevant knowledge.

Such time limitation is inadequate to study the application of different models and techniques, which requires a more experiment-oriented training, since most theoretical results available (e.g. worst case complexity) have limited application. For this purpose, practical experimentation with simple problems is done in the laboratory classes, and is assessed in more complex projects, requiring the students to develop analysis and decision skills of great relevance in this area.

3.3.9. Bibliografia principal:

Livros de Texto / Text Books

- Jaulin, L., Kieffer, M., Didrit, O., Walter, E., Applied Interval Analysis, Springer, 2001*
- Jorge Cruz, Constraint Reasoning for Differential Models, 126 Frontiers in Artificial Intelligence and Applications, IOS Press, 2005.*
- Francisco Azevedo, Constraint Solving over Multivalued Logics - Application to Digital Circuits, 91 Frontiers of Artificial Intelligence and Applications, IOS Press, 2003.*

3.3.1. Unidade curricular:

Computação de Alto Desempenho / High Performance Computing

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Abílio Duarte de Medeiros (T-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Maria Cecília Farias Lorga Gomes (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

A unidade curricular está no bloco de especialização e tem como principal objetivo fornecer competências na área de High Performance Computing, ou seja estudar as metodologias e técnicas que permitem explorar arquiteturas hardware com múltiplos processadores. O objetivo é diminuir os tempos de execução de programas com necessidade de recursos de computação elevados. O curso também comporta uma vertente aplicada e experimental com uma atividade laboratorial. Supõe-se que os estudantes conhecem conceitos básicos de arquitetura, redes e sistemas de operação.

Saber:

- *Arquiteturas de multiprocessadores*
- *Técnicas de decomposição de problemas*
- *Linguagens, bibliotecas e ferramentas necessárias para as várias fases de desenvolvimento de um programa paralelo*

Saber Fazer:

- *Otimização do desempenho de um programa sequencial*
- *Utilização de ferramentas de medida de desempenho de multiprocessadores e de clusters*
- *Programação de multiprocessadores usando C/C++, OpenMP, MPI e OpenCL*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The course is included in the specialization part of the degree and its main goal is to give competences in the area of High Performance Computing; i.e. to the methodologies and techniques that allow the exploitation of hardware architectures with multiple processors, in order to reduce the execution times of programs that need high computational resources. The course also takes a hands-on approach with laboratory and project activity. Students should have knowledge about computer architecture, computer networks and operating systems

Knowledge:

- *Multiprocessor architectures*
- *Techniques for problem decomposition*
- *Languages, libraries and tools needed for all the development cycle of a parallel program*

Application:

- *Performance optimization of a sequential program*
- *Ability to use tools for measuring the performance of symmetric multiprocessors and clusters*
- *Multiprocessor programming using C/C++ OpenMP, MPI and OpenCL*

3.3.5. Conteúdos programáticos:

1. *Otimização de programas sequenciais. Técnicas e ferramentas relevantes.*
2. *Multiprocessadores de memória partilhada e clusters*
3. *Programação de multiprocessadores de memória partilhada. OpenMP*
4. *Programação de clusters. MPI. Ferramentas para exploração de clusters.*
5. *Operações de entrada e saída e sistemas de ficheiros orientados para alto débito. MPI-IO.*
6. *Computação usando GPUs. OpenCL.*
7. *Algoritmos e Aplicações: algoritmos de ordenação, álgebra linear, processamento de imagem, procura e otimização*

3.3.5. Syllabus:

1. *Performance optimization of sequential programs*
2. *Shared memory multiprocessors and clusters*
3. *Programming shared-memory multiprocessors. OpenMP.*
4. *Programming distributed-memory multiprocessors. MPI. Cluster tools.*
5. *High throughput input/output and file systems. MPI-IO.*
6. *GPU computing. OpenCL.*
7. *Algorithms and applications: sorting algorithms, linear algebra, image processing, search and optimization*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O objetivo principal de fornecer competências para o uso de multiprocessadores é atingido pela conjugação das duas partes da cadeira: a primeira em que se percorrem as metodologias e as ferramentas necessárias às diferentes fases do ciclo de vida de um programa paralelo; na segunda esses conhecimentos são exercitados e consolidados através da análise de várias áreas de aplicação em que o potencial para aplicação da computação paralela é enorme, incluindo a motivação para a necessidade do desenvolvimento de algoritmos que podem ter pouco a ver com os algoritmos sequenciais aplicados nas mesmas áreas.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The course aims to give competences for exploiting multiprocessors; this objective is achieved through the combination of the two parts of the course: in the first part, students are exposed to the methodologies and tools used in the life cycle of a parallel program; in the second part, this knowledge is consolidated by applying it to different areas where there is a great potential for parallel computing use. From this second part, contributions are made towards the idea that parallel algorithms can be very different from the sequential ones used in the same area.

3.3.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas têm por objetivo a apresentação dos principais temas e discutir as questões mais relevantes. As sessões laboratoriais têm lugar num laboratório de uso geral com acesso a PCs (que são multicore) quer para a programação de memória quer para a de memória distribuída. Um cluster dedicado é também utilizado na disciplina, permitindo o acesso a ferramentas de gestão e monitorização que não estão disponíveis no laboratório de uso geral.

Componentes da avaliação:

- *Dois testes intermédios (com o peso total de 60% na nota final)*
- *Dois projetos de programação (com o peso total de 40% na nota final)*

3.3.7. Teaching methodologies (including evaluation):

Lectures are intended to support the instructor's presentation of fundamental issues.

Laboratory sessions take place in a general-purpose laboratory with PC-based multicore nodes for shared memory and distributed-memory programming. A dedicated cluster is also available where cluster management and monitoring tools are installed.

Assessment Components:

- *Two intermediate tests (60 % of the final grade)*
- *Two programming assignments (40% of the final grade)*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Como atrás indicado, as duas partes da cadeira contribuem para os objetivos da aprendizagem. Para qualquer das partes, a compreensão e o estudo dos tópicos principais é objeto das aulas teóricas e avaliado através de testes.

Quanto às sessões laboratoriais, eles são de natureza diferente nas duas partes. Na primeira parte, as sessões laboratoriais consistem na resolução de problemas de pequena dimensão que permitem experimentação com bibliotecas e ferramentas que implementam os conceitos expostos. Na segunda parte, as aulas práticas são usadas para a discussão das soluções a usar nos projetos de programação; as técnicas e os conceitos expostos na 1ª parte da cadeira são revisitados e o seu uso contribui de forma fundamental para serem atingidos os objetivos de aprendizagem.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

As explained before, both parts of the course contribute to the learning objectives. Comprehension of the main topics is dealt by lectures and evaluated with tests.

Laboratory sessions have different characteristics in the two parts. In the first part, practical classes consist of the use of libraries and tools that implement the methodologies and techniques exposed in the corresponding lectures. In the course part related with applications and algorithms for selected areas, laboratory sessions are dedicated to discussion of particular aspects of the solution development. The concepts and techniques of the

first part of the course are revisited as their use makes a fundamental contribution to the fulfilment of the learning objectives,

3.3.9. Bibliografia principal:

Course support book

• *B. Wilkinson, M. Allen, "Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers (2nd Edition)", Prentice-Hall 2004*

Other books

• *P. Pacheco, "An Introduction to Parallel Programming", Morgan Kaufman, 2011*

• *G. Hager, G. Wellein, "Introduction to High Performance Computing for Engineers and Scientists", CRC Press, 2011*

• *B. Gaster, L. Howes, D. Kaeli and P. Mistry, "Heterogeneous Computing with OpenCL", Morgan Kaufman, 2011*

Mapa IV - Computação Multimédia / Multimedia Computation

3.3.1. Unidade curricular:

Computação Multimédia / Multimedia Computation

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Nuno Manuel Robalo Correia (T-28h; P-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

N/A

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *As características principais dos diferentes tipos de informação.*
- *Os mecanismos de estruturação mais relevantes.*
- *Os paradigmas de autoria existentes.*
- *As características das bibliotecas de suporte para construção de aplicações multimédia.*
- *Algoritmos de processamento de sinal.*

Saber Fazer:

- *Escolher o formato mais apropriado para um determinado tipo de informação.*
- *Definir a estrutura subjacente a uma aplicação multimédia.*
- *Programar bibliotecas para análise, processamento e visualização.*
- *Programar algoritmos de processamento.*
- *Programar aplicações multimédia.*

Competências Complementares:

- *Entender o carácter multidisciplinar da disciplina e a relação com outras áreas.*
- *Capacidade para gerir e desenvolver parcialmente um projeto multimédia.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *The main characteristics of the different types of information.*
- *The most important structuring mechanisms.*
- *The most common authoring paradigms.*
- *The characteristics of the support libraries for building multimedia applications.*
- *Signal processing algorithms.*

Application:

- *Choose the most appropriate format for a particular type of multimedia information.*
- *Define the architecture of a multimedia application.*
- *Program libraries for analysis, processing and visualization.*
- *Program processing algorithms.*
- *Program multimedia applications.*

Soft-skills:

- *Understand the multidisciplinary nature of the area and the relationship with other areas.*
- *Ability to manage and partially develop a multimedia project.*

3.3.5. Conteúdos programáticos:

1. *Informação estática e informação dinâmica*
 - 1.1. *Texto, imagens, áudio, vídeo*
 - 1.2. *Formatos e normas de representação, compressão e transmissão*
2. *Modelos de estruturação: Hipertexto, hipermédia, hipervídeo*
 - 2.1. *Conceitos: Estrutura e navegação*
 - 2.2. *Modelos e arquiteturas de sistemas hipermédia*
 - 2.3. *Hipermédia na Internet*
 - 2.4. *Tempo e Hipermédia*
3. *Ferramentas de autoria*
 - 3.1.1. *Paradigmas*
 - 3.1.2. *Exemplos práticos*
4. *Arquiteturas de software e modelos de objetos para multimédia*
5. *Análise, processamento e visualização de informação*
6. *Pesquisa, recuperação e reutilização*
7. *Perspetivas futuras em computação multimédia*

3.3.5. Syllabus:

1. *Static and dynamic information*
 - 1.1. *Text, images, audio, video*
 - 1.2. *Formats and standards of representation, compression and transmission*
2. *Structuring models: Hypertext, hypermedia, hypervideo*
 - 2.1. *Concepts: Structure and Navigation*
 - 2.2. *Models and hypermedia systems architecture*
 - 2.3. *Hypermedia and the Web*
 - 2.4. *Time and Hypermedia*
3. *Authoring tools*
 - 3.1. *Paradigms*
 - 3.2. *Examples*
4. *Software architectures and object models for multimedia*
5. *Information analysis, processing and visualization*
6. *Search, retrieval and repurposing*
7. *Perspectives on multimedia computing*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O programa inclui o processo e os métodos para a construção de sistemas e aplicações multimédia. São abordados os diferentes tipos de informação multimédia e a sua representação computacional. Analisam-se os paradigmas dominantes para estruturação de informação. Estudam-se e experimentam-se bibliotecas de suporte à programação, no âmbito de aplicações que envolvem manipulação e processamento de informação. É apresentada e discutida investigação recente na área e são analisadas as perspetivas futuras.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The program includes the process and methods for the development of multimedia systems and applications. The different types of multimedia information and its computational representation are addressed. The dominant paradigms for structuring information are presented. In the context of applications that involve multimedia information processing, support libraries are studied and experimented. Recent research in the field is presented and discussed and future perspectives are analyzed.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas é apresentada a matéria, com exemplos e referências a sistemas relacionados. As aulas laboratoriais destinam-se à realização do projeto que acompanha a matéria apresentada na aulas teóricas. A avaliação da disciplina inclui cinco elementos: dois testes escritos individuais efetuados ao longo do semestre e 1 projeto, com várias fases (especificação, artigo com trabalho relacionado e código/interface), que correspondem a um projeto de uma aplicação multimédia. Fórmula de cálculo da nota final: Nota_final = 25% Teste1 + 25% Teste2 + 10% Especificação + 15% Artigo + 25% Código & Interface. A aprovação à disciplina exige as seguintes notas:

- (média (teste_escrito1; teste_escrito2) >= 10) AND*
- (média (Especificação; Artigo; Código & Interface) >= 10) AND*
- (Teste_escrito1 >= 8) AND*
- (Teste_escrito2 >= 8)*

Os alunos que obtiveram notas positivas no trabalho prático e não obtiveram aprovação nos testes poderão realizar um exame, cuja nota substituirá a nota dos testes no cálculo da nota final.

3.3.7. Teaching methodologies (including evaluation):

In the lectures the course content is presented, illustrated with application examples and references to related work. The laboratory classes are intended for specification, development and presentation of the project that deals with topics presented during the lectures. The evaluation of the course consists of five elements: two mid-term written individual tests and one project, with several phases (specification, state of the art and code/interface) that together account for a project to develop a multimedia application, using information processing. Final grade formula:

Final_Grade = 25%Test1+25%Test2+10%Specification+15%Paper+25%Code&Interface

Course approval requires the following minimal grades:

- (mean (Test1;Test2) >= 10) AND*
- (mean(Specification;Paper;Code&Interface) >= 10) AND*
- (Test1 >= 8) AND*
- (Test2 >= 8)*

Students approved in the project and not approved in the tests may also do a final exam, and the resulting grade replaces the grades of the tests in the final grade formula.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas teóricas são expostos os princípios fundamentais da unidade curricular, acompanhados de exemplos e de referências para trabalho relacionado. Nas aulas práticas o alunos desenvolvem um projeto de dimensão média, que acompanha os tópicos da componente teórica.

Para além das bibliografia recomendada, os alunos têm acesso aos slides apresentados nas aulas teóricas e a artigos científicos, de forma a enquadrar os tópicos e o trabalho no contexto mais genérico dos sistemas multimédia atuais.

Durante as aulas práticas, os alunos têm que trabalhar em grupos (preferencialmente de dois elementos) e aplicar os conhecimentos proporcionados pelas aulas teóricas. O projeto tem uma dimensão apreciável e vários momentos de verificação, de forma a ser semelhante a um trabalho de equipa real, no desenvolvimento de um sistema multimédia.

Para obter aprovação, o aluno deve assistir a pelo menos dois terços das aulas práticas. Os testes escritos individuais permitem a avaliação mais precisa dos conhecimentos adquiridos por cada aluno.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The fundamental principles are presented during the lectures, illustrated with examples and references to related work. In the laboratory classes the students develop a project, that implements the topics of the theoretical component.

In addition to the recommended readings, students have access to the slides presented in the lectures and related scientific, to contextualize the topics and the project in the context of current multimedia systems.

During the laboratory classes, students have to work in groups (preferably with two elements) and apply the knowledge obtained from the lectures and other education materials. The project is relatively large with several deliverables to replicate as much as possible the situation of a real team developing a multimedia system.

To obtain approval, the student must attend at least two thirds of the practical classes. The individual written tests allow a more accurate assessment of knowledge acquired by each student.

3.3.9. Bibliografia principal:

Text Book references:

- Multimédia e Tecnologias Interativas, Nuno Ribeiro, FCA, 2004 (2ª Edição, 2006).*
- Tecnologias de Compressão Multimédia, Nuno Ribeiro e José Torres, FCA, 2009.*
- Fundamentals of Multimedia, Ze-Niam Li, Mark S. Drew, Pearson, Prentice Hall, 2004.*
- Programming Interactivity, Joshua Noble, O'Reilly Media, 2009.*

Additional readings will be provided during classes and through the course website (including lecture slides).

Mapa IV - Conceitos e Tecnologias XML / Concepts and Technologies of XML

3.3.1. Unidade curricular:

Conceitos e Tecnologias XML / Concepts and Technologies of XML

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:
Carlos Augusto Isaac Piló Viegas Damásio (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:
João Carlos Gomes Moura Pires (Regente) (T-28h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Conceitos fundamentais de XML (espaços de nomes, documentos bem formados e válidos, modelo hierárquico de dados).*
- *O papel das diferentes tecnologias de validação de documentos, de transformação, de descrição e de interrogação.*
- *Entender a definição de um vocabulário XML e regras de boa marcação. Entender definições em DTDs ou um XML Schemas.*
- *A filosofia declarativa do XSLT, a estrutura de caminhos do XPath e a forma funcional de XQuery.*

Saber Fazer:

- *Definir um vocabulário para um fim específico e o seu DTD e/ou XML Schema.*
- *Processar XML (para XML ou HTML) usando XSLT ou qualquer das APIs.*
- *Instalar e configurar os servidores necessários a uma solução Web.*
- *Implementar uma solução com um cliente Web, recorrendo às tecnologias XML.*
- *Organizar e interrogar um repositório de documentos XML.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Basic XML concepts (name-spaces, well-formed documents, valid documents, hierarchical data model, data-oriented versus text oriented XML documents, etc.)*
- *The role of each of the XML associated technologies such that for validation, description, transforming and processing, querying, linking*
- *The basics for a XML vocabulary definition and the available technologies like DTD, XML schema*
- *The philosophical principle of separation of concerns from content, logic and presentation*
- *The declarative nature of XSLT for XML processing; The declarative and functional nature of XQuery*

Application:

- *To define an XML language for a specific application and to create a DTD or XML Schema.*
- *To process XML documents using XSLT and at least one of the standard APIs*
- *To select, install and configure the servers to build a web based solution*
- *To implement a small/medium size web base application very configurable, using XML technologies*
- *Design and explore an XML data repository*

3.3.5. Conteúdos programáticos:

1. Introdução ao XML

- 1.1. Motivações e características de desenho do SGML, HTML, XML**
- 1.2. Regras sintáticas XML. Entidades. Regras de boa marcação. Estilos de Marcação.**
- 1.3. Vocabulários XML para documentos e para dados. Exemplos e diferenças.**
- 1.4. Modelo do documento XML e XPath.**
- 1.5. XPath.**
- 1.6. Panorâmica das tecnologias XML e o papel e status de cada uma delas.**
- 1.7. Separação entre conteúdo e apresentação. Arquiteturas**

2. Validação de Documentos

2.1. DTD

2.2. XML Schema

2.3. Outras abordagens

3. Transformação de Documentos

3.1. XSLT

3.2. Outras abordagens

3.3. O papel do XSLT:FO

4. Base de Dados XML

4.1. Panorâmica e princípios

4.2. XQuery

4.3. XUpdate e XML:DB

5. APIs para trabalhar com documentos XML

- 5.1. DOM
- 5.2. SAX
- 6. Metadados e XML

3.3.5. Syllabus:

- 1. Introduction
 - 1.1. Motivations and design options on SGML, HTML, XML
 - 1.2. XML syntactic rules. Entities. Rules for a good markup. Markup styles.
 - 1.3. XML vocabularies for text and for data. Examples and differences.
 - 1.4. XML model document and XPath
 - 1.5. Overview of XML technologies, their roles and status.
 - 1.6. Fundamental approach: separating the format from the content.
- 2. Validation of XML documents
 - 2.1. DTD
 - 2.2. XML Schema
 - 2.3. Other approaches.
- 3. Transformation and processing of XML documents
 - 3.1. XSLT
 - 3.2. Other approaches
 - 3.3. The role of XSL:FO
- 4. XML Databases
 - 4.1. Overview and principles
 - 4.2. XQuery
 - 4.3. XUpdate and XML:DB
- 5. XML APIs
 - 5.1. DOM
 - 5.2. SAX
- 6. Metadata and XML

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Esta unidade curricular apresenta uma panorâmica atualizada das tecnologias XML, e da sua utilização nomeadamente na área de sistemas de Informação e da Web.

É dado um especial realce á tecnologias seguintes: extração de informação de um documento usando Xpath; definição de vocabulários XML e validação de documentos usando DTD, XML Schema, e adicionalmente Schematron; transformação de documentos XML usando XSLT; interrogação de bases de dados XML usando XQuery.

É ainda apresentado o RDF e RDF/XML. Mapeamento de e para modelos relacionais. Aplicação de algumas APIS de entre, SAX, DOM.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The course on XML Concepts and Technologies presents a broad and updated overview of XML and its associated technologies and their role on the building of modern Information Systems.

A particular emphasis is given to the following topics: information extraction from XML documents using XPath and XQuery; definition of XML vocabularies and validation using DTD, XML Schema, Schematron; transformation of XML documents using XSLT. The RDF and RDF/Schema are also introduced in the last section. The use of standard APIs (DOM, JDOM, SAX, etc.) is exercised.

Information Systems architectures based on XML technologies are discussed and introduced, namely the role and the bridge with relational databases

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas e práticas. Nas primeiras são lecionados os conceitos e tecnologias relevantes. Nas aulas práticas são resolvidos problemas com o objetivo de os alunos desenvolverem as competências básicas nas diferentes tecnologias. Adicionalmente, nas aulas práticas serão iniciados alguns trabalhos que contribuem para a componente de avaliação.

A avaliação de conhecimentos inclui:

- *1 teste individual teórico, onde é avaliado o conhecimento que os alunos adquiriram dos conceitos e características das tecnologias lecionadas.*
 - *2 trabalhos práticos, realizados por grupos de dois alunos, que incidem sobre a problemas que requerem o uso de conhecimentos de parte da UC.*
 - *1 trabalho prático, realizados por grupos de dois alunos, que integra os diferentes conhecimentos da UC.*
- O exame de recurso substitui a teste individual.*

3.3.7. Teaching methodologies (including evaluation):

The syllabus is taught in theoretical and laboratory classes. In the former, the main concepts and techniques are addressed. The lab sessions are used to solve small problems for basic training. Some lab sessions are used to setup their team works used for evaluation purpose.

The evaluation of students include:

- *1 theoretical individual test, where the knowledge acquired by the students on the concepts and technologies are assessed.*
- *2 practical projects, for 2 students teams, that evaluate their competence on partial subjects.*
- *1 practical project, for 2 students teams, that evaluate their competence on a project that integrates many competences.*

The final exam replaces the theoretical individual test.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As linguagens de marcas oferecem uma representação flexível e estruturada da informação e tem permitido uma melhor integração entre o mundo dos documentos e o mundo dos dados. O XML e as tecnologias associadas constituem um conjunto de conceitos, modelos, standards e tecnologias que tem permitido a aplicação de linguagens de marcas a todos os domínios da atividade humana quer para a definição de modelos e formatos para interoperação entre sistemas, quer para representação e criação de repositórios de informação estruturada ou semi estruturada.

A aquisição de competências para a definição, extensão, manipulação, transformação e interrogação de documentos numa ou mais linguagens de marcas requer um forte domínio tecnológico que é adquirido pelas aulas práticas e pelos dois tipos de trabalhos práticas incluídos na avaliação. Por outro lado a rápida evolução tecnológica sugere que, mais do que uma constante atualização do conteúdo ou dos trabalhos, é fundamental numa aquisição e domínio dos conceitos fundamentais, que é salvaguardada pelas aulas teóricas durante todo o semestre e num teste teórico individual que irá cobrir toda os assuntos discutidos na UC.

O terceiro trabalho desenvolve e avalia a capacidade de desenho de uma pequeno sistema de informação com uma forte componente em documentos XML com interface e integração na Web.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The markup languages offer a flexible but structured representation of information and that has enabled an easier integration of data and documents worlds. The XML (which is a meta-markup language) and the associated technologies embody a set of concepts, models, standards and tools that have been applied to all domains.

The competences for definition, extension, processing and query documents in one or more markup languages require strong technological skills, which are addressed in the lab sessions and through the two types of team-works. On the other hand, the fast technological evolution suggest that just making a regular program update is not enough, because a deep understanding of the concepts and models behind the technology is essential, which is provided by the lectures and the individual theoretical test.

The third practical (team work) project develops and evaluates the skills to design a small information system with a strong component on XML documents fully integrated with the Web. and with a suitable interface.

3.3.9. Bibliografia principal:

- *Elliott Rusty Harold, W. Scott Means, XML in a Nutshell, Third Edition, Third edition, 2004, ISBN: 0596007647.*
- *Dmitry Kirsanov, XSLT 2.0 Web Development (Charles F Goldfarb Definitive Xml Series), Prentice Hall PTR (March 25, 2004), ISBN: 0131406353.*
- *Howard Katz et al., XQuery from the Experts: A Guide to the W3C XML Query Language, Addison-Wesley Professional, 2004, ISBN: 0321180607.*
- *Gavin Powell, Beginning XML Databases (Wrox Beginning Guides), John Wiley & Sons (14 Nov 2006), ISBN: 0471791202*

Mapa IV - Data Warehousing / Data Warehousing

3.3.1. Unidade curricular:

Data Warehousing / Data Warehousing

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Manuel Corrêa Calvente de Barahona (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Carlos Gomes Moura Pires (Regente) (T-28h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *As arquiteturas, componentes e processos em Data Warehousing.*
- *O que é a análise de multidimensional dados (OLAP) e como se insere em Sistemas de Apoio à decisão.*
- *Aspetos principais do modelo multidimensional e operações típicas.*
- *Meta modelos para modelos multidimensionais.*
- *Desenho e interrogação de um modelo ROLAP.*
- *Conceitos base em S(patial)OLAP.*

Saber Fazer:

- *Análise e desenho de modelos multidimensionais.*
- *Análise e desenho dos processos de ETL.*
- *Exploração analítica de um modelo multidimensional.*
- *Análise preliminar de dimensionamento e taxa de crescimento do modelo. Análise preliminar de desempenho.*
- *Derivação de um modelo multidimensional a partir de uma modelo OLTP.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Data Warehouse architectures, components and processes.*
- *What is OLAP and what is its role in Decision Support Systems?*
- *Multidimensional Model main characteristics and their typical operations.*
- *Meta-models for multidimensional models.*
- *Design and querying ROLAP models.*
- *Basic concepts of S(patial)OLAP.*

Application:

- *Analysis and design multidimensional models*
- *Analysis and design ETL processes.*
- *Analytical exploitation of a multidimensional model*
- *Space and performance analysis of multidimensional models.*
- *Deriving multidimensional models from an OLTP model.*

3.3.5. Conteúdos programáticos:

- 1. Introdução ao Data Warehousing*
- 2. Introdução ao OLAP*
 - 2.1. Modelo multidimensional*
 - 2.2. Operações base*
 - 2.3. Interação na atividade de análise de dados*
 - 2.4. Visualização da informação*
- 3. Data Warehouse segundo a escola E/R*
 - 3.1. Subject Data Model; Business Data Model, System Model; Technology Model*
 - 3.2. Tópicos avançados de modelação: Histórico, Hierarquias, Transações*
- 4. Data Warehouse segundo a escola multidimensional*
- 5. Arquitetura*
- 6. Tópicos avançados de modelação multidimensional*
- 7. Derivação do modelos analíticos a partir de modelos E/R*
- 8. Modelos conceptuais Multidimensional*
- 9. Fundamentos do ETL*
- 10. Fundamentos para implementação física*
- 11. Integração de Informação geográfica em sistemas OLAP*
- 12. Estudo de casos*

3.3.5. Syllabus:

- 1. Introduction to Data Warehousing*
- 2. Introduction to OLAP*
 - 2.1. Multidimensional model*
 - 2.2. Typical OLAP operations*
 - 2.3. Interaction in the data analysis process*

2.4. Information visualization

3. Data Warehouse according to the E/R school

3.1. Subject Data Model; Business Data Model, System Model; Technology Model

3.2. Advanced topics: historical data, hierarchical data, Transactions data

4. Data warehouse according to the multidimensional school

4.1. Architecture

4.2. Advanced topics on multidimensional modeling

5. Deriving analytical models from E/R models

6. Conceptual multidimensional data models

7. Introduction to ETL.

8. Introduction to physical implementation

9. Integration of geographic information into OLAP

10. Case studies

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

É apresentada uma panorâmica atualizada dos sistemas de apoio à decisão baseados em dados, com foco em Data Warehouse, OLAP, modelos e modelação multidimensional.

As arquiteturas, componentes e processos em Data Warehousing são apresentadas e discutidos os detalhes de cada uma das escolas (Imnon e Kimball) nas secções de ETL e processos físicos.

Os conceitos, modelos e atividades em OLAP, são explicados nas teóricas e explorados nas atividades recomendadas. Em complemento, são discutidos o impacto e as oportunidades abertas ao OLAP pela introdução de informação geográfica nos modelos multidimensionais.

O programa discute extensivamente aspetos de modelação multidimensional, com uma secção específica para o desenho do modelo mutidimensional, quando este deriva de um único OLTP.

A modelação discutida na secção de tópicos avançados de modelação multidimensional, inclui os aspetos de análise do crescimento e do desempenho dos modelos, nomeadamente ROLAP.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course offers an updated overview of data driven decision support systems with a focus on Data Warehouse, OLAP, multidimensional models and modelling.

The Data Warehouse's architectures, components and processes are presented and the details of the main schools (Imnon e Kimball) discussed. Additionally the sections on ETL and physical implementation complement the essential on DW.

Very early in the syllabus, a attention is drawn to the main OLAP concepts, models and activities. These aspects are complemented, at the end, with a discussion on the use of geographical information in multidimensional models.

It is worth to notice that a significant part of the syllabus is concerned with multidimensional modelling and includes a specific section when the multidimensional model is derived from a single OLTP system.

The design of multidimensional models in the section "Advanced topics on multidimensional modelling" includes aspects of analysis of space growing and performance.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas onde são apresentados os conceitos, complementadas com aulas práticas onde se abordam artigos, e são seguidos alguns tutoriais para familiarização com alguns conceitos ou tecnologias.

A avaliação de conhecimentos inclui:

- 1 trabalho prático, para grupos de 2 alunos, para familiarização e aplicação de alguns conceitos base (20%).*
- 1 trabalho prático, para grupos de 2 alunos, para especificação do âmbito, arquitetura e o meta-modelo a adotar num trabalho prático a implementar posteriormente. (20%)*
- 1 trabalho prático, para grupos de 2 alunos, com a implementação do trabalho anteriormente especificado, integrando diferentes conhecimentos da UC. (35%)*
- 1 teste individual teórico, onde é avaliado o conhecimento que os alunos adquiriram dos conceitos teóricos apresentados e discutidos nas aulas, e das leituras e atividades recomendadas (25%).*

O exame de recurso substitui o teste individual.

3.3.7. Teaching methodologies (including evaluation):

The syllabus is taught in theoretical classes addressing the main concepts and techniques, complemented with lab classes with recommended readings and activities, namely following tutorials to get insights on some concepts or technologies.

The evaluation of students include:

- 1 practical work, for 2 students teams, to get contact with some basic concepts and their application (20%).*

- *1 practical project, for 2 students teams, that should specify the scope, the architecture and the meta-model to be adopted in the project to be implemented (20%).*
- *1 practical project, for 2 students teams, that should implement the previously specify project and has to be discussed (35%).*
- *1 theoretical individual test, where is assessed the knowledge acquired by the students on the concepts presented and discussed during the lectures, and the recommended readings and activities (25%).*
The final exam replaces the theoretical individual test.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Apesar da área de Data Warehouse se caracterizar por uma forte componente tecnológica, não existe propriamente um conjunto de tecnologias padrão sendo muito diversas as ferramentas na área de ETL, exploração, análise e visualização de dados, etc.

Assim, o foco é essencialmente nos conceitos e na sua aplicação e não numa ou noutra tecnologia específica. Deste modo existem apenas aulas teóricas complementadas por um conjunto de leituras e aulas práticas com atividades recomendadas que devem permitir consolidar os conceitos ao longo do semestre.

O modelo de referência para as componentes OLAP é o ROLAP o que permite, com a bagagem anterior em SQL, que os alunos desenvolvam uma compreensão, não apenas dos diferentes tipos de modelos multidimensionais, como também da sua exploração analítica e a sua tradução em termos de interrogação. Permite igualmente explorar de forma adequada diferentes conceitos tais como preagregados, dimensões compatíveis, a definição de meta-modelos e o seu mapeamento físico (ROLAP), entre outros. e melhor explorar alguns tópicos avançados de modelação multidimensional.

O método de avaliação, decomposto em quatro componentes, contribui quer para a consolidação dos conhecimentos quer para o desenvolvimento de algumas competências. O primeiro trabalho prático permite quer a familiarização quer a experimentação com as operações base do OLAP. Os dois outros trabalhos permitem a consolidação de muitos dos conceitos trabalhados durante o semestre através da sua aplicação. O teste individual permite avaliar o conhecimento de cada um dos alunos sobre todos os tópicos desenvolvidos durante o semestre.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Data Warehouse is a strongly technological area without widely adopted technological standards, rather with a large number of different tools and approaches to the some task. Even the efforts to define a metadata standard have not reach a clear success. However, at conceptual level, a body of knowledge has to be understood even considering two main schools (Imnon and Kimball).

The teaching methodologies will thus focus on concepts and their application, instead of addressing a particular technology or tool. Consequently the course has no lab sessions and lectures will be used to present and discuss the main concepts, complemented with the recommended readings and activities to consolidated the concepts during the semester.

The used referenced model for OLAP is the ROLAP. With the student previous SQL knowledge, it is possible and easy to develop a better understanding of the different multidimensional models and their analytical exploration and querying. It is also an excellent framework to explore concepts like aggregates, conform dimensions, meta-models and their corresponding physical mappings. It is also an appropriate reference to discuss advanced multidimensional modelling topics.

The evaluation method, with four components, will contribute to their knowledge consolidation and to the development of some skills. The first practical work will provide a familiarization and experimentation with the basic of analytical exploration in OLAP. The others two practical works will provide a deep understanding of some of the concepts presented during the semester, by applying them in a project. The individual test assesses the students' knowledge about all the topics covered during the semester.

3.3.9. Bibliografia principal:

- Ralph Kimball, Laura Reeves, Margy Ross, Warren Thornthwaite, The Data Warehouse Lifecycle Toolkit : Expert Methods for Designing, Developing, and Deploying Data Warehouses., Wiley, 1998.*
- Ralph Kimball, Margy Ross. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition). Wiley, 2002.*
- W. H. Inmon. Building the Data Warehouse (3rd Edition). Wiley. (4rd Edition), 2005.*
- Claudia Imhoff, Nicholas Gallemmo, Jonathan G. Geiger. Mastering Data Warehouse Design: Relational and Dimensional Techniques. Wiley, 2003.*
- Joe Caserta, Ralph Kimball. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning. Wiley, 2004.*

Complementary readings:

- From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design -*

from Daniel L. Moody.

-Hierarchies in a Multidimensional Model: From conceptual Modeling to Logical Representation from E. Malinowski, E. Zimányi.

Mapa IV - Gestão de Centros de Dados / Data Center Management

3.3.1. Unidade curricular:

Gestão de Centros de Dados / Data Center Management

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Abílio Duarte de Medeiros (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Paulo Orlando Reis Afonso Lopes (Regente) (T-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Fazer:

- Centros de Dados (CD): caracterização e classes de serviços
- Avaliar e planear níveis de serviço, quanto ao desempenho e disponibilidade
- Identificar e avaliar arquiteturas HW e SW que suportam os diferentes níveis de serviço
- Identificar/avaliar as ferramentas disponíveis para o administrador de sistemas
- Identificar e planear soluções que minimizem o consumo de energia

Saber Fazer:

- Afetação de recursos ("capacity planning") para um nível de serviço
- Construir soluções HW/SW garantindo níveis definidos de escalabilidade e tolerância a faltas
- Monitorizar e administrar os recursos do centro de dados desde a infraestrutura às aplicações e serviços
- Estabelecer as políticas de "backups" e de "disaster recovery"
- Desenvolvimento das capacidades de comunicação oral e escrita com diferentes públicos-alvo (gestores, fornecedores, técnicos de TI)

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- Data Centres (DC): characterization and service classes
- Assess and plan service levels towards performance and availability
- Identify and evaluate HW and SW architectures that meet designated service levels
- Identify/evaluate datacenter management tools
- Identify and plan for lower energy consumptions (green datacenter)

Application:

- Capacity planning to meet a designated service level
- Design HW/SW solutions that meet designated scalability and fault tolerance targets
- Monitor and manage DC resources, from infrastructure to applications and services
- Define backup and disaster recovery policies
- Develop written and oral presentation skills targeting distinct audiences (management, suppliers, IT staff)

3.3.5. Conteúdos programáticos:

- Arquiteturas software: monolíticas a multitier. Adaptar, interoperar, manter e evoluir.
- Arquiteturas de sistemas computacionais: Mainframe, SMP, NUMA, clusters e farms. CPU, memória e subsistema de I/O. Infraestruturas de rede LAN; Armazenamento e sistemas de ficheiros. Armazenamento distribuído (blocos e ficheiros). Convergência de infraestruturas de rede.
- Isolamento, consolidação e virtualização. Particionamento. Contentores. Tipos de virtualização. Cloud computing. Sistema computacional elástico. Virtualizar tudo: processador, memória, rede e armazenamento. Modelos de serviço oferecidos na cloud.
- Desenho da arquitetura. Desempenho, escalabilidade e disponibilidade. Arquiteturas geograficamente distribuídas.
- Monitorar e analisar: modelos, ferramentas e benchmarks.
- Gestão do centro de dados. Políticas, metodologias e ferramentas. Lidar com a heterogeneidade. Cópias de salvaguarda e recuperação. Disaster recovery.

•O centro de dados: organização, energia e arrefecimento

3.3.5. Syllabus:

- Software architectures: monolithic to multitier. Adapt, interoperate, maintain and evolve. Performance, scalability and fault-tolerance.*
- Computer system architectures: Mainframe, SMP, NUMA, clusters and farms. The CPU, memory and I/O subsystems. Network infrastructures: LAN and low latency technologies. Storage and File Systems. Distributed block and file storage. Converged network infrastructures.*
- Isolation, consolidation and virtualization. Partitioning. Containers. The virtualization spectrum. Cloud computing. The elastic, pay-per-use computer system. Virtualize everything: processor, memory, network, storage. Cloud service models.*
- Architecture design. Scale, performance and availability. Geographically distributed architectures.*
- Monitoring and analysis: Models, tools and benchmarks.*
- Data centre management. Policies, methodologies and tools. Dealing with heterogeneous HW and SW. Backup and recovery. Disaster recovery.*
- Data centre room: Organization, power and cooling*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Apresentam-se as tecnologias base e arquiteturas – computacionais, de armazenamento e de interconexão – que se encontram em centros de dados avançados. A necessidade de soluções de alta disponibilidade e elevado desempenho são motivo para introduzir os aspetos de tolerância a faltas, escalabilidade, clusters e farms. Estes aspetos são apresentados tanto sob o prisma da arquitetura da aplicação e respetivo middleware de suporte, como das tecnologias necessárias para os realizar. A otimização de recursos é o pano de fundo para a apresentação do particionamento, virtualização, e cloud. São também apresentadas e discutidas as questões de monitorização, de desempenho, e gestão operacional do CD. As sessões (tanto seminários como sessões laboratoriais, visitas de campo e/ou demonstrações) conduzidas por convidados que representam empresas líderes nas TI reforçam os aspetos “avançado” e “prático”, contribuindo para cumprir os objetivos desta unidade curricular.

3.3.6. Demonstration of the syllabus coherence with the curricular unit’s intended learning outcomes:

Fundamental technologies and architectures - compute, storage and interconnection - used in advanced data centers are introduced. The need for high availability and high performance solutions are used to drive the aspects of fault tolerance, scalability, clusters and farms. These aspects are presented both from the aspect of application architecture and its middleware support, as the technologies needed to achieve them. The optimization of resources is the motivation for the presentation of partitioning, virtualization, and cloud computing. Monitoring, performance, and operational management of the Data Centre are also discussed. The sessions (both seminars and lab work, field visits and/or demonstrations) conducted by invited speakers of IT leading companies reinforce the "advanced" and "practical" aspects, helping to fulfill the objectives of this course.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é desenvolvido através de sessões teóricas conduzidas pelo docente da disciplina, e seminários, sessões práticas laboratoriais e/ou de demonstração, e visitas de campo conduzidas por convidados. A avaliação assenta em testes presenciais individuais e em relatórios técnicos (surveys) e/ou trabalhos laboratoriais ao longo do semestre, em grupos.

3.3.7. Teaching methodologies (including evaluation):

The course is based both on theoretical sessions led by the academic staff, and seminars, practical sessions and/or demonstrations, and field visits led by invited speakers and leading IT companies. The assessment is based on individual classroom tests and technical reports (surveys) and/or group-based laboratory work, during the semester.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As sessões teóricas conduzidas pelo docente da disciplina servem para introduzir os conceitos, tecnologias e componentes essenciais, bem como a sua utilização para construir arquiteturas com provas dadas; por outro lado, os seminários conduzidos pelos convidados apresentam aos alunos o “estado na arte” na realização destas mesmas tecnologias e componentes.

As sessões práticas laboratoriais e/ou de demonstração conduzidas pelos convidados permitem aos alunos

contactar/ utilizar/ trabalhar com tecnologias, ferramentas e produtos software cujo licenciamento e necessidade de infraestruturas dedicadas e de “grande” escala tornam impossível a sua existência nos laboratórios da universidade.

Finalmente, as visitas de campo, conduzidas por convidados, a grandes Centros de Dados em Portugal possibilitam aos alunos o contacto que, mais uma vez, só é possível fora da universidade.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit’s intended learning outcomes:

Lectures delivered by the instructor introduce concepts, technologies and key components as well as their use to build proven architectures, while invited speakers’ seminars present “state of the art” implementation of these technologies and components.

The practical lab sessions and /or demonstration conducted by guests allow students to contact / use /work with technology, tools and software products whose licensing and needs for “large” dedicated infrastructures make it impossible to exist in the university labs.

Finally, field visits, conducted by guests, to large data centers in Portugal allow students to experience something that, once again, is only possible outside the university.

3.3.9. Bibliografia principal:

Capítulos seleccionados de/Selected chapters:

•K.Jayaswal,Administering Data Centers:Servers, Storage,and Voice over IP,Wiley,2006

•J.Allspaw, The Art of Capacity Planning ,O’Reilly, 2008

•Mark Burgess,Analytical Network and System Administration: Managing Human-Computer System,John Wiley and Sons, 2004

•D.Menascé, V.Almeida, L.Dowdy, Performance by Design: Computer Capacity Planning by Example, Prentice Hall, 2004

•R.Chevancé,Server Architectures:Multiprocessors,Clusters,Parallel Architectures, WEB Servers and Storage Solutions, Elsevier, 2005

Also, selected papers, technical reports, white papers, product manuals and other industry or scientific information publicly available. As a short example, we cite:

•Steve C.Chiu, Alok N.Choudhary, Danli Wang. Network and device-level impacts: performance and reliability of active I/O storage systems. In J.Supercomputing (2007) 41:163–178

•Special issue of the ACM SIGOPS Operating System Review on “Systems work at VMware”.44-4, Dec 2010

Mapa IV - Gestão de Projetos Informáticos / IT Project Management

3.3.1. Unidade curricular:

Gestão de Projetos Informáticos / IT Project Management

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Manuel Corrêa Calvente de Barahona (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Francisco José Massa de Menezes (Regente) (TP-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- Conceitos de gestão de projetos

- Especificidades dos projetos informáticos

- Técnicas de planeamento e estimação

- Técnicas de comunicação oral e escrita

- Técnicas de gestão de pessoas e equipas

Saber Fazer:

- Definir o âmbito e os requisitos do projeto

- Desenvolver um cronograma de um projeto (CPM/GANTT),

- Utilizar uma ferramenta de planeamento

- Definir um modelo de gestão de custos do projeto

- Elaborar um plano de gestão dos riscos

- Elaborar um plano de gestão da comunicação no projeto

Competências Complementares:

- *Aptidões de avaliação*
- *Aptidões de organização do trabalho*
- *Gestão do tempo e cumprimento de prazos*
- *Trabalho em equipa e aptidões de cooperação*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Project management principles*
- *Planning and estimating techniques*
- *Oral and written communication techniques*
- *Techniques for team management*

Application:

- *Defining project requirements and scope*
- *Developing a project schedule (CPM/GANNT)*
- *Using planning tools*
- *Defining a project cost model*
- *Defining a risk management plan*
- *Developing a project communications plan*

Soft-skills:

- *Evaluation skills*
- *Planning and organizing skills*
- *Time management*
- *Team work and cooperation skills*

3.3.5. Conteúdos programáticos:

1. Conceitos de gestão de projetos. Especificidades dos projetos informáticos.

1.1 Processos da gestão de projetos

1.2 Ciclo de vida de um projeto.

1.3 Modelos de desenvolvimento de projetos de software

1.4 Project management knowledge áreas (PMBOK Guide Fourth Edition)

2. Organização e planeamento do projeto

2.1 Definição do âmbito e dos requisitos do projeto

2.2 Cronograma do projeto (técnica CPM). Uso do MS Project Professional

2.3 Recursos e custo do projeto

2.4 Plano de gestão do risco

2.5 Controlo das alterações ao projeto

3. Monitorização e controlo do projeto

3.1 Gestão da configuração

3.2 Medida e controlo do progresso

3.3.5. Syllabus:

1. Project management concepts and framework. Information Systems projects.

1.1 Project management processes

1.2 Project life cycle

1.3 Software project development models

1.4 Project management knowledge areas (PMBOK Guide Fourth Edition)

2. Project planning and organization

2.1 Definition of project requirements and project scope

2.2 Project schedule (CPM technique). us of MS Project Professional

2.3 Project resources and cost estimation

2.4 Risk management plan

2.5 Project change control

3. Project monitoring & control

3.1 Configuration management

3.2 Progress control

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Nesta unidade curricular cobrem-se os aspectos da gestão de projectos, mais relevantes para a inserção com sucesso do aluno na realidade Profissional de projectos informáticos.

Nesse sentido, a linha condutora será de revelar os conceitos e cenários mais úteis na execução de projectos

Informáticos, ao longo de um ciclo completo de um projecto para se consolidarem os objectivos propostos, com ênfase na simulação de situações mais típicas de um projecto real.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course addresses the most relevant aspects of Project management, for the successful insertion of the student in the real world of IT projects.

In that sense, the guiding line is to reveal useful concepts and scenarios for the execution of IT projects, following

a full project life cycle to consolidate the proposed objectives, focusing on the most typical situations of a real life

project.

3.3.7. Metodologias de ensino (avaliação incluída):

- *Apresentação de conceitos em aulas teóricas a serem avaliados em 2 testes escritos;*
- *Execução na prática e em grupo de um ciclo completo de gestão de projectos, com aplicação contínua dos conceitos teóricos e experimentação de ferramentas de planeamento;*
- *O projecto a executar terá dois milestones com avaliação de resultados através da apresentação dos entregáveis pedidos e da exposição oral do estado dos projectos;*
- *Investigação e discussão de temas/ferramentas com os alunos.*

3.3.7. Teaching methodologies (including evaluation):

- *Presentation of concepts in the theory classes, to be evaluated in 2 written tests;*
- *Practical execution in a team of a full life cycle Project, continuously applying the theoretical concepts and experimenting planning tools;*
- *The team project will have two milestones with the evaluation of the requested deliverables and oral exposition of the project status;*
- *Research and discussion of concepts/tools with the students.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A metodologia empregue, ao aplicar conceitos de gestão de projectos a um caso prático completo, expõe o aluno à realidade de um projecto informático dotando-o de skills essenciais ao sucesso desde os seus primeiros

projectos profissionais. O objectivo é o de preencher os gaps mais importantes para quem enceta uma actividade

profissional como consultor de IT.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The adopted methodology exposes the student to the professional context of IT projects, allowing him to acquire essential skills to the immediate success in his/her future activity, filling important gaps to a full blown professional IT consultant.

3.3.9. Bibliografia principal:

- António Miguel. Gestão de Projetos de Software, 4ª edição atualizada, FCA, 2010.*
- Bob Hughes, Mike Cotterell. Software Project Management, 5th edition, McGraw-Hill Higher Education, 2009.*
- Mike Cohn. Agile Estimating and Planning, Prentice Hall, 2008.*

Mapa IV - Linguagens Para Domínios Específicos / Domain Specific Languages

3.3.1. Unidade curricular:

Linguagens Para Domínios Específicos / Domain Specific Languages

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Ana Maria Diniz Moreira (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Vasco Miguel Moreira do Amaral (Regente) (T-28h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- a) Entender riscos e oportunidades de DSL versus GPL*
- b) Conhecer o processo completo de Engenharia de Linguagens*
- c) Aprender técnicas de Análise de Domínio*
- d) Usar abordagens Orientadas a Modelos para desenvolvimento de linguagens*
 - Simulação*
 - Execução*
 - Análise (Verificação e Validação)*
- e) Aprender ferramentas formais e práticas (“workbenches”) de desenvolvimento de DSLs relacionando com conhecimentos adquiridos noutras disciplinas*
- f) Conhecer exemplos de domínio de aplicação de DSLs*
- g) Aprender a avaliar uma linguagem do ponto de vista da usabilidade*
- h) Conhecer as vias de investigação correntes sobre DSLs*

Fazer:

- i) Usar ferramentas de desenvolvimento de DSLs*
- j) Dado um problema num domínio específico, saber como desenvolver uma DSL de raiz*

Competências Complementares:

- k) Conhecer o processo de revisão de artigos numa conferência (tanto na ótica de autor como de revisor)*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- a) Understand the risks and opportunities of developing a DSL over GPL solution*
- b) Know the Software Languages Engineering process*
- c) Learn techniques of Domain Analysis*
- d) Effective use of Model-Driven Development for language development*
 - o Simulation*
 - o Execution*
 - o Analysis*
 - o Model-based Testing*
- e) Learn formal and practical tools (workbenches) for developing DSLs relating this knowledge with the one previously acquired in other classes*
- f) Learn several typical domains of application of DSLs*
- g) Know how to evaluate a language from the point of view of usability*
- h) Follow and understand the up-to-date developments of DSLs in the research community*

Application:

- i) Know how to use up-to-date technology (tools) to develop DSLs*
- j) Given a domain-specific problem, know how to implement a DSL from scratch*

Soft-skills:

- k) Know the paper reviewing process for a conference, both in the perspective of the author and the reviewer*

3.3.5. Conteúdos programáticos:

1. Motivação para estudo das LDEs

- Definição de LDE*
- Vantagens Vs. Desvantagens*
- Comparação LDEs Vs. Linguagens de Propósito Geral*
- Textual Vs. Visual*

2. Engenharia de Linguagens

- Processo de desenvolvimento*
- Engenharia de Domínio*
- Desenvolvimento de Software Orientado a Modelos (OM) aplicado a linguagens*
- Abordagem formal*

3. Modelos e MetaModelos de Linguagens

- Sintaxe Concreta e Abstrata*
- Frases bem formadas (impondo restrições com OCL)*
- Semântica de Linguagens: Domínios Semânticos e Mapeamentos*
- Formalismos*

4. Transformação e Composição de Modelos

- Tipos de Linguagens de transformações e sua aplicação*

- Fundamento: Gramáticas de grafos
- Transformações de Alto Nível
- 5. Propósitos pragmáticos para o uso das linguagens
- Execução Computacional
- Análise de Modelos e sua Verificação
- Testes baseados em Modelos
- Simulação
- 6. Outras abordagens não OM para implementação de LDEs
- 7. Avaliação e Validação experimental de LDEs
- 8. Casos de Estudo em Domínios diferentes

3.3.5. Syllabus:

1. Motivation to study DSL
 - Definition of DSL
 - Advantages Vs. Disadvantages of DSL
 - Comparing DSL Vs. GPL (General Purpose Language)
 - Textual Vs. Visual Languages
2. Language Engineering
 - Development Process
 - Domain Engineering
 - Model-Driven Software Development (MDD)
 - Formal Approach
3. Language Models and MetaModels
 - Abstract and Concrete Syntax
 - Well-formedness (OCL constraints)
 - Language Semantics: Semantic Domains and Semantic Mappings
 - Formalisms
4. Model Transformation and Composition
 - Types of Transformations and Transformation Languages (their purpose)
 - Foundations: Graph Grammars
 - High-Order Transformations
5. Language pragmatic purposes
 - Execution
 - Model Analysis and Verification
 - Model-Based System Testing
 - Simulation
6. Other non-MDD approaches to language development
7. DSL evaluation and validation
8. Case Studies in different domains

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Uma LDE oferece poder expressivo com nível de abstração adequado ao utilizador final, sendo orientada para um domínio específico (e.g. engenharia de software; algoritmos criptográficos; configuração de sistemas; bases de dados; multimédia; telecomunicações; bio-informática; simulação; CAD; controle de equipamentos; análise de dados; processamentos financeiros; sistemas Ciber-Físicos).

A UC aborda metodologias e ferramentas de desenvolvimento de LDEs baseadas em Desenvolvimento de Software Orientado a Modelos (MDD). Os objetivos:

- a) e b) são apresentados no conteúdo temático 1, e c) discutido no ponto 2. Ambos serão avaliados sob a forma de teste.
- d) é detalhado em 3, 4 e 5, com práticas em laboratórios e avaliados sob a forma de projeto (P1 e P2).
- e) e também i) e j), são trabalhados em aulas de laboratório, sendo usados e avaliados no projeto (P1 e P2)
- f) é discutido em 7.
- g) e h) são usados em 7 (projeto). A avaliação da workshop (W) inclui um pequeno estado da arte.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

A DSL offers expressive power oriented to a specific domain, at the right level of abstraction to the operator (e.g. software engineering; Cryptography algorithms; system management; data bases; multimedia; telecommunications; bio-informatics; simulation; CAD; hardware control; data mining; financial computations; cyberphysical systems).

This course will approach methodologies and tools for developing a DSL, based on Model-Driven Development techniques. Following the list of bullets presented before:

- a) and b) are presented in 1 and c) is discussed in 2. Both will be evaluated by a test.

- d) will be deeply discussed in 3., 4. and 5. Besides being practiced in the lab and workgroup project (P1 and P2).
- e) It will be exercised in the lab, similar to i) and j), besides the project (P1 and P2)
- f) will be discussed in 7.
- g) and h) will be discussed and exercised both in the lab and project 7 in the written paper for the workshop (W). It should include a state-of-the-art.

3.3.7. Metodologias de ensino (avaliação incluída):

Aulas teóricas (uma aula semanal de duração de 2 horas) consistem na exposição de matéria, ilustrada com exemplos de aplicação. As aulas de laboratório (uma aula de 2 horas semanais) serão dedicadas ao uso de ferramentas, acompanhado de diversas ilustrações e exercícios.

O aluno terá de construir com sucesso um projeto em grupo, incluindo: 1) Construção da DSL - análise de domínio, proposta de desenho da linguagem adequada ao problema e sua implementação; 2) Transformação – Plataformas alvo, simulação, execução e análise. Artigo de investigação com apresentação oral numa sessão de palestras destinadas à apresentação dos trabalhos no fim do semestre com a crítica a artigos dos colegas e apresentações orais. A nota final calculada de acordo com a expressão:

*Nota Final = 0,25 * P_DSL + 0,30 * P_Transformations + 0,20 * Workshop + 0,30 * Test*

Acesso ao exame de recurso depende do desempenho satisfatório com média entre 9.5 em 20.

3.3.7. Teaching methodologies (including evaluation):

Lectures (hours weekly) present the topic and illustrate them with application examples. Lab classes (2 hours weekly) dedicated to the use of tools, followed by several illustrations and exercises (e.g. Language Workbenches for Eclipse, generation of editors for visual or textual languages; Tools for Model-Transformations; Verification with ALPINA and/or Formula .

The students will have to successfully build a project in two distinct evaluation phases (DSL design and implementation + Transformation and Target Platforms). The evaluation is composed by this Project (55% of the total grade) in group work, a research assignment (paper to be presented at the workshop) and one test by the end of the semester.

*Final Grade = 0,25 * P_DSL + 0,30 * P_Transformation + 0,20 * Workshop + 0,30 * TEST*

Access to the ressit exam is conditional upon the satisfactory performance with a grade (computed out of the sum of all componentes) of at least 9,5 out of 20.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os conceitos introdutórios de LDEs são apresentados aos alunos através de exemplos práticos e com uso de formalismos e ferramentas.

Para a maturação dos conceitos, é realizado um conjunto de exercícios nas aulas práticas onde os alunos deverão aplicar os conhecimentos adquiridos para além de praticarem com ferramentas em ambiente Eclipse para suporte à Engenharia de Linguagens como por exemplo: GMF/EMF, Eugenia (para geração de editores de linguagens visuais); EMF ou XText (para geração de editores de linguagens textuais); ATL, DSLtrans (para transformações de modelo-para-modelo); Xpand (para transformações modelo-texto); ALPINA (Model Checker); Formula (Constraint Solver).

Os alunos vão sendo avaliados ao longo do semestre através do desenvolvimento de um trabalho prático com dois entregáveis (em anos anteriores consistiu: no desenvolvimento de uma DSL para Legos MindStorm para execução e Simulação num ambiente 3D; no desenvolvimento de uma DSL para jogos Role-Play com execução gráfica e Verificação em Alpina) e por um teste no final do semestre.

Há lugar a apresentação oral por parte do grupo numa workshop de um artigo sobre temas relacionados com a cadeira a definir. Será exercitada a análise crítica (os alunos serão revisores dos artigos dos colegas) simulando um ambiente comum numa conferência. O mapeamento das relações com os objetivos já são discutidos nas secções anteriores.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Introductory concepts of DSLs are presented to the students by means of practical examples and use of formalisms and tools.

In order to mature the concepts, the student will follow several exercises at the Lab. There, the students should apply their acquired knowledge and practice with tools in Eclipse to support the Software Language Engineering activity. Examples of tools are: GMF/EMF, Eugenia (generating editors for visual languages); EMF or XText (for generating editors for textual languages); ATL, DSLtrans (for Model-To-Model Transformations); Xpand (For Model-to-Code Transformations); and tools like ALPINA and Formula for Verification purposes.

The students are evaluated along the semester by means of the project with several deliverables (in previous years consisted of: developing a DSL for Lego MindStorms with execution, and simulation on a 3D environment; Developing a DSL for Role-Playing games with visual execution, and automatic Verification of properties using

Alpina) and a testa t the end of the semester.

The project evaluation ends with an oral presentation at a internal workshop, together with a report written like a research paper. By reviewing each others' publications, the students will exercise their soft skills like critics. The direct mapping of the methodologies with the goals, were already discussed in the previous sections.

3.3.9. Bibliografia principal:

Text Book:

-A.Kleppe, Software Language Engineering: Creating Domain-Specific Languages Using Metamodel, Addison-Wesley, 2008

-Steven Kelly, J. Tolvanen, Domain-Specific Modeling, IEEE Wiley, 2008

-M. Utting and B. Legeard, Practical Model-Based Testing: A Tools Approach Morgan-Kaufmann, 2007

-K. Czarnecki and U. Eisenecker, Generative Programming Methods, Tools and Applications, Addison-Wesley, 2000

-J. Greenfield and K. Short, Software Factories, Wiley Publishing, 2004

-G. Rozenberg, Handbook of Graph Grammars and Computing by Graph Transformation,1, World Scientific Pub Co Inc, 1997

- B. Berard et al, Systems and Software Verification - Model-Checking Techniques and Tools, Springer,1999

- A. Barisik, V. Amaral, M. Goulão, B. Barroca, Evaluating the usability of Domain-Specific Languages, in Formal ad Practical Aspects of Domain-Specific Language: recent developments, IGI Global, 2012

Complementary materials:

- Coleção de slides da cadeira. Folhas de exercícios propostos

Mapa IV - Lógicas para a Programação e Especificação / Programming and Specification Logics

3.3.1. Unidade curricular:

Lógicas para a Programação e Especificação / Programming and Specification Logics

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (T-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

António Maria Lobo César Alarcão Ravara (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber: Esta unidade ensina conceitos e técnicas avançadas, baseados em lógica construtiva, em que se apoia a produção de software certificado e que suportam o raciocínio rigoroso sobre programas e linguagens de programação. A relação próxima entre programas e demonstrações / provas, chamada a correspondência de Curry-Howard, será revelada e explorada na unidade. Os estudantes conhecerão os modernos fundamentos da teoria de tipos da programação funcional e imperativa, incluindo a programação com tipos dependentes, assim como o desenvolvimento integrado de provas de correção.

Fazer: Os estudantes aprendem a desenvolver programas certificados e tipos de dados usando Coq, um assistente de prova de nível industrial, desenvolvido no INRIA, e usado internacionalmente na indústria e academia. Exemplos do tipo de software coberto inclui programas funcionais e imperativos baseados em ADTs, e interpretadores e processadores de linguagens, incluindo verificadores de tipos e compiladores.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge: This course teaches advanced concepts and techniques, based on constructive logic, which underpin the production of certified software and support reasoning about programs and programming languages.

The close connection between programs and proofs, called the Curry-Howard correspondence, will be revealed and explored in the course. Students learn the modern type-theoretical foundations of functional and imperative programming, including programming with dependent types, and to develop program proofs.

Application: Students also how to develop certified programs and data types using Coq, an industrial strenght proof assistant, developed at INRIA, being used worldwide in industry and academia. Students use a modern interactive theorem proof assistant. Examples of covered software include ADT based functional and imperative programs, programming language interpreters, and processors, including type checkers and compilers.

3.3.5. Conteúdos programáticos:

1. Lógica Construtiva

Proposições como Tipos/Programas como Provas; A interpretação Curry-Howard; A lógica do Coq.

2. Raciocínio sobre Programas Funcionais com Tipos Dependentes

Programas funcionais básicos: listas, ADTs, provas básicas; Tipos dependentes; Tipos indutivos; Propriedades Indutivas; Polimorfismo e funções de ordem superior. Extração de Programas. Táticas em Coq.

3. Raciocínio sobre Interpretadores de Programs

Representação de Expressões e Comandos; Especificação da Avaliação: Triplos de Hoare e precondições mais fracas, condições de verificação; Raciocínio sobre programas imperativos que manipulam estruturas de dados.

4. Raciocínio sobre Linguagens de Programação

Linguagens funcional simples; Sistemas de tipos; Preservação de Tipos; Linguagem máquina simples; Especificação da Execução; Correção de Compilador. Equivalência; Bisimulação.

NB: Todos os capítulos também cobrem noções relevantes do Coq, tais como manipulação de demonstrações, táticas e automatização de provas.

3.3.5. Syllabus:

1. Constructive Logic

Propositions as Types / Programs as Proofs; The Curry-Howard Interpretation; Logic in Coq.

2. Reasoning about Functional Programs with Dependent Types

Basic Functional Programming: numbers, lists, basic proofs; Dependent types; Inductive types; Inductive propositions; Polymorphism and Higher-order functions. Program extraction. Tactics in Coq.

3. Reasoning about Program Interpreters

Representing Expressions and Commands; Specification of Evaluation: Hoare Triples, weakest preconditions, verification conditions; Reasoning about imperative programs that manipulate data-structures.

4. Reasoning about Programming Languages

A Simple Functional Language; Specification of Evaluation; Type Systems; Type Preservation and Type Safety; A Simple Machine Language; Specification of Execution; Compiler correctness. Program Equivalence; Bisimulation.

NB: All chapters will also cover relevant Coq notions, such as proof manipulation, tactics, and proof automation.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Existe uma correspondência evidente entre os conteúdos e os objetivos.

Um princípio pedagógico da UC é a apresentação coesa da teoria com a sua aplicação prática, neste caso com recurso ao assistente de prova Coq. Esta organização de conteúdos programáticos já foi testada pelo menos numa edição da unidade, e segue a estrutura de unidades semelhantes a nível internacional.

Os objetivos de “Saber” são cobertos nos tópicos 1, 2, 3 e 4, com ênfase em 1, 2.

Os objetivos de “Fazer” são cobertos nos tópicos 1, 2, e 3 e 4 com ênfase em 2, 3, 4.

3.3.6. Demonstration of the syllabus coherence with the curricular unit’s intended learning outcomes:

There is an evident correspondence between the syllabus and the curricular unit’s objectives.

A pedagogical principle enforced in the course is the tightly coupled presentation of theory with application, in this case resorting to the intensive use of the Coq proof assistant. The organization of topics was already tested in at least one edition of the curricular unit, and follows the structure of similar courses offered at the international level.

The “Knowledge” learning outcomes are covered in items 1, 2, 3 e 4, with an emphasis in 1, 2 .

The “Application” learning outcomes are covered in items 1, 2, e 3 and 4, with emphasis in 2, 3, 4.

3.3.7. Metodologias de ensino (avaliação incluída):

O ensino está organizado em aulas teóricas (2T) e práticas (2P). Nas aulas teóricas são apresentados os conceitos e discutidas situações problemáticas. Tipicamente, algumas das competências de saber fazer são também exercitadas nas aulas teóricas, de forma a aumentar a ligação entre os conceitos teóricos e a sua aplicação. Nas aulas práticas os alunos discutem e resolvem exercícios propostos pelo docente, usando o assistente de prova Coq de forma intensiva. A avaliação contínua consiste em dois testes (um intermédio, outro no final do semestre) e 1 projeto de desenvolvimento de software, realizado em grupo. São também apresentados 4 pequenos trabalhos, que deverão ser resolvidos nas aulas laboratoriais (uso de ferramentas). Os alunos que tenham reprovado na avaliação contínua com uma nota igual ou superior a 8 podem tentar fazer o exame final, cuja nota combinada com a dos trabalhos práticos produz a nota final.

3.3.7. Teaching methodologies (including evaluation):

The course is organized in recitation lectures (2T) and laboratory classes (2P).

In the laboratory classes students discuss and solve problems proposed by the instructor from a predefined list, and using intensively the Coq proof assistant. In the recitation lectures the instructor presents and motivates concepts and applications are discussed and exemplified. Typically, some “knowledge application” learning outcomes are also exercised in the recitation, so to promote a close connection between the theoretical concepts and their application. Evaluation consists in midterm and final tests and one software development project, to be done by the students as a team. Any student that fails in the basic evaluation scheme with a grade equal of higher than 8/20 is admitted to the final exam, which combines with the grading of the project to produce the final grade.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os trabalhos práticos e o projeto de grupo permitem prender os alunos a um forte ritmo de acompanhamento da matéria. O uso de ferramentas de demonstração assistida Coq promove a motivação dos alunos para a adoção de técnicas rigorosas de verificação de software, já que torna muito claro o poder das mesmas no aumento de confiança do software produzido. A formação teórica, que contribui para a compreensão profunda dos mecanismos de funcionamento das ferramentas também reforça os objetivos de aprendizagem. A metodologia seguida, apelando à intuição e esclarecendo bem a utilidade dos resultados teóricos em cada passo, produz em geral bons resultados.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The laboratory work assignments and the team project contributes to bind the students while following the course presentation. The use of the proof assistant Coq facilitates the motivation of students towards the adoption of rigorous verification techniques, since they are able to clearly see the empowering effect on the robustness of the software produced, and the impact of logical reasoning on program validation. The study of theory, contributing for the deep understanding of the working principles of tools, also enforces the learning outcomes. The methodology we have devised, which appeals to intuition, and clarifies at each step the usefulness of the presented theoretical results, seems to provide good results.

3.3.9. Bibliografia principal:

*“Software Foundations”, Benjamin C. Pierce, Chris Casinghino, Michael Greenberg, Vilhelm Sjöberg, Brent Yorgey, 2012. The Coq Proof Assistant (<http://coq.inria.fr/>).
Lecture Notes and Hand-outs, Luís Caires, 2012.*

Mapa IV - Modelos de Concorrência e Segurança / Models of Concurrency and Security

3.3.1. Unidade curricular:

Modelos de Concorrência e Segurança / Models of Concurrency and Security

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

*Carla Maria Gonçalves Ferreira (Regente) (T-14h; P-14h)
António Maria Lobo César Alarcão Ravara (T-14h; P-14h)*

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- as linguagens de modelação de sistemas concorrentes baseadas em comunicação CCS, cálculo- π , e Applied- π ;*
- noções de equivalência comportamental entre sistemas concorrentes, reconfiguráveis e criptográficos;*
- especificar propriedades comportamentais e temporais simples (justiça, não-bloqueio, não-interferência, etc.) dos sistemas considerados;*
- expressar propriedades de segurança (confidencialidade, autenticidade, controle de acessos) usando modelos operacionais e lógicos.*

Fazer:

- aplicar as linguagens de modelação apresentadas em contextos de aplicação concretos (algoritmos concorrentes, distribuídos, protocolos de segurança);*
- exprimir propriedades de um sistema em termos de propriedades de um modelo;*
- representar um protocolo de segurança definido informalmente numa especificação verificável, detetando e resolvendo ambiguidades;*
- usar ferramentas de verificação de sistemas e protocolos de segurança.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- Modelling languages for communication based concurrent systems as CCS, pi calculus, and applied-pi;*
- Behaviour equivalence between concurrent, reconfigurable, and cryptographic systems;*
- Specifying simple behavioural and temporal properties (fairness, non-blocking, non-interference, etc.);*
- Express security properties (confidentiality, authenticity, access control) through logic and operational models.*

Applications:

- Apply the learned modelling languages to concrete applications (concurrent and distributed algorithms, security protocols);*
- Express system properties in terms of a model properties;*
- Representing an informally defined security protocol as a verifiable specification, in order to verify and solve ambiguities;*
- Use verification tools for security systems and protocols.*

3.3.5. Conteúdos programáticos:

1. Concorrência e comunicação:

-modelação de sistemas concorrentes baseados em comunicação: sistemas de transição etiquetados; CCS (sintaxe, semântica operacional, equivalência de comportamentos); verificação de propriedades por equivalência entre descrições concretas e abstratas de sistemas

-especificação de propriedades de sistemas: lógica de Hennessy-Milner e lógicas temporais (LTL e CTL)*

-verificação com o TAPAS

2. Mobilidade e Segurança:

-modelação de sistemas reconfiguráveis: cálculo-pi (sintaxe, semânticas operacionais e equivalências)

-especificação de propriedades de sistemas: lógica espacial (sintaxe, semântica, padrões de especificação)

-modelação e verificação de propriedades com equivalências (no MWB) e lógica espacial (no SLMC)

-modelação de sistemas criptográficos: Applied-pi (sintaxe, semântica operacional e equivalência)

-especificação e análise de propriedades de segurança de sistemas no modelo de Dolev-Yao (autenticidade, confidencialidade e confiança), com o Proverif

3.3.5. Syllabus:

1. Concurrency and communication:

Modelling communication based concurrent systems: labelled transition systems; CCS (syntax, semantics, behavioural equivalence); verification of properties through equivalence between concrete and abstract system descriptions;

Specification of system's properties: Hennessy-Milner logic and temporal logic (LTL and CTL)*

Verification with TAPAS

2. Mobility and security:

Modelling reconfigurable systems: pi calculus (syntax, semantics, and equivalence)

Specification of system's properties: spatial logic (syntax, semantics, and specification patterns)

Specification and verification of properties with equivalences (with MWB) and spatial logic (with SLMC)

Modelling cryptographic systems: applied-pi (syntax, semantics, and equivalence)

Specification and analysis of security properties in the Dolev-Yao model (authenticity, confidentiality, and trust), with Proverif.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Saber linguagens de modelação de sistemas concorrentes: estuda-se o CCS, o cálculo-pi e o Applied-pi. Aplicar as linguagens na modelação de sistemas concretos: especificam-se pequenos exemplos reais.

Saber noções de equivalência comportamental entre sistemas: para cada linguagem de modelação estudam-se equivalências. Aplicar essas noções na verificação de propriedades: testa-se a equivalência entre descrições concretas e abstratas de sistemas com ferramentas computacionais (TAPAS, MWB).

Saber especificar propriedades comportamentais de sistemas concorrentes, usando lógicas temporais e espaciais: estuda-se HML, LTL, CTL e SL, com ênfase na especificação. Aplicar as lógicas na verificação:*

validam-se propriedades de sistemas concretos, com as ferramentas (TAPAS, SLMC). Saber exprimir propriedades de segurança, com modelos operacionais e lógicos: faz-se a modelação destas propriedades e de sistemas criptográficos. Aplicar os conceitos na verificação: usa-se a ferramenta Proverif.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

To know modelling languages for communication based concurrent systems: this course presents CCS, pi calculus and applied pi. These modelling languages are applied to small real systems.

To understand behaviour equivalence between systems: behavioural equivalences are studied for each modelling language discussed. Those equivalences are applied for the verification of properties: TAPAS and MWB tools are used to test equivalence between concrete and abstract description of a system.

To know how to specifying behavioural and temporal properties: this course presents HML, LTL, CTL, and SL, emphasising specification of properties. Applying logics to the verification of properties: some properties of concrete systems are validated with TAPAS and SLMC.*

To know how to express security properties through logic and operational models: this course addresses how to model cryptographic systems security properties.

3.3.7. Metodologias de ensino (avaliação incluída):

A metodologia seguida consiste na exposição da matéria em aulas teóricas, ilustrada com casos de estudo, e suportada pela utilização de ferramentas de verificação. As aulas laboratoriais consistem na resolução de exercícios, nos quais os aluno terá que modelar sistemas, verificar bissimilaridade entre modelos, especificar e verificar propriedades sobre modelos.

A avaliação é composta por dois testes e dois trabalhos individuais. Ambos os trabalhos consistem na modelação de um sistema concorrente (os trabalhos diferem no nível de complexidade do sistema a modelar), na especificação e verificação de um conjunto de propriedades e na elaboração de um relatório.

Condições para obter aprovação:

NE \geq 9.5 e NT \geq 9.5,

em que NE é a média aritmética das notas dos testes e NT é a nota do trabalho prático.

3.3.7. Teaching methodologies (including evaluation):

Lectures include theoretical exposition, illustrated by case studies, and supported by the use of verification tools. In laboratory sessions, students solve exercises: modelling systems, verifying bissimilarity between models, specifying and verifying model's properties.

Assessment includes two tests and two individual practical assignments. In both of the assignments, the student will have to model a concurrent system (these assessments differ on the complexity of the system being model), specify and verify a set of properties and write a report.

Conditions for approval:

NE \geq 9.5 and NT \geq 9.5,

where NE is the mean of tests grades and NT is the practical assignment grade.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas teóricas são apresentadas as linguagens de especificação dos sistemas e das suas propriedades (sintaxe, semântica e noções de equivalência), ilustrando-as com pequenos exemplos reais e propriedades genéricas. Nos testes procura-se avaliar se a/o aluna/o é capaz de identificar a melhor linguagem para modelar dado sistema ou propriedade, se consegue usar a linguagem para representar o sistema ou propriedade em questão, e se conhece e sabe aplicar os resultados apresentados na teórica para verificar propriedades de sistemas.

Nas aulas práticas tratam-se pequenos sistemas aplicativos reais (tais como protocolos): faz-se a sua modelação a partir de descrições informais, especificam-se e verificam-se as propriedades básicas que devem ser respeitadas ("à mão" e usando ferramentas computacionais). No primeiro trabalho prático modela-se um sistema concorrente simples. É usado o CCS e verificam-se não só equivalências comportamentais como propriedades lógicas temporais, usando a ferramenta TAPAS. No segundo trabalho prático modela-se um sistema concorrente reconfigurável com requisitos de segurança. É usado o cálculo-pi e o Applied-pi e verificam-se não só propriedades lógicas como se segurança, usando as ferramentas MWB e Proverif. A componente teórica da avaliação testa o domínio dos conceitos, seus contextos de aplicação e suas limitações. Os alunos devem compreender o papel, capacidade expressiva e a necessidade das várias linguagens e as implicações na sua semântica e nas noções de equivalência. Devem também entender os padrões de especificação de propriedades e como instancia-los para as várias linguagens estudadas. A componente prática da avaliação testa se os alunos são capazes de usar as linguagens para representar sistemas reais e verificar as propriedades que estes devem ter.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In lectures consist on presenting the specification languages and their properties (syntax, semantics, and equivalences), illustrating those languages with small real examples and generic properties. Tests assess the student's ability to identify the best-suited language to model a specific system or property, and if the student knows and is able to apply the results for the verification of systems discussed in the lectures.

Lab sessions deal with small realistic systems (as protocols): modelling those systems from an informal description, specifying and verifying basic properties (both on paper and with tools). In the first practical assignment, students will have to model with CCS a simple concurrent system. Furthermore, students will use the TAPAS tool to prove behavioural equivalences and temporal properties. In the second practical assignment, students will model with pi calculus and applied calculus a reconfigurable concurrent system with security requirements. In this last assessment, students will use MWB and Proverif for verifying not only logic properties but also security properties.

Tests assess the student's knowledge of the concepts addressed, their application contexts and limitations. Also, for each language addressed, students should understand, its role, its expressiveness and applicability, equivalence notions and the implications of its semantic model. Learn the patterns for the specification of properties and how to instantiate them in the languages studied.

The individual practical assessments check if the students are able to use the specification languages to model and verify properties of real systems.

3.3.9. Bibliografia principal:

- Robin Milner, *Communication and Concurrency*, Prentice Hall 1989
- Davide Sangiorgi and David Walker, *The Pi-Calculus - a theory of mobile processes*, Cambridge University Press 2001
- Luca Aceto, Anna Ingólfssdóttir, Kim Guldstrand Larsen, and Jiri Srba, *Reactive Systems: Modelling, Specification and Verification*, Cambridge University Press 2007
- Michael Huth and Mark Ryan, *Logic in Computer Science: modelling and reasoning about systems*, Cambridge University Press 2004
- Mark D. Ryan and Ben Smyth, *Applied pi calculus*, in V. Cortier & S. Kremer (eds), *Formal Models and Techniques for Analyzing Security Protocols*, Chapter 6, IOS Press 2011

Mapa IV - Pesquisa e Otimização / Search and Optimization

3.3.1. Unidade curricular:

Pesquisa e Otimização / Search and Optimization

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Pedro Manuel Corrêa Calvente de Barahona (T-28h; P-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

N/A

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Problemas de satisfação de restrições e de otimização.*
- *Tipos de restrições (simples / globais).*
- *Propagação de restrições e tipos de consistência.*
- *Algoritmos completos e incompletos para satisfação/otimização.*
- *Heurísticas e meta-heurísticas.*

Saber Fazer:

- *Modelação de problemas de satisfação/otimização de restrições.*
- *Escolha das restrições mais adequadas.*
- *Escolha dos algoritmos mais apropriados.*
- *Escolha das heurísticas mais eficientes.*

Competências Complementares:

- *Compreensão de especificações informais (eventualmente incompletas).*
- *Identificação de problemas de restrições.*
- *Análise e explicação de resultados.*
- *Capacidade de pesquisa de literatura.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Constraint satisfaction and optimisation problems.*
- *Types of constraints (simple / global).*
- *Constraint propagations and types of consistency.*
- *Complete and incomplete algorithms for satisfaction/optimisation.*
- *Heuristics and meta-heuristics.*

Application:

- *Modelling of constraint satisfaction/optimisation problems.*
- *Selection of most adequate constraints.*
- *Selection of most appropriate algorithms.*
- *Selection of most efficient heuristics.*

Soft-skills:

- *Understanding of (possibly incomplete) informal specifications.*
- *Identification of constraint satisfaction/optimisation problems.*
- *Analysis and explanation of results*
- *Specialised literature searching*

3.3.5. Conteúdos programáticos:

1.Problemas de Decisão e Otimização

1.1.Exemplos

1.2.Complexidade.

2.Resolução Completa de Restrições - Pesquisa com Retrocesso

2.1.Tipos de restrições

2.1.1.Aritméticas

2.1.2.Disjuntivas

2.1.3.Reificadas

2.1.4.Globais

2.2.Redes de restrições

2.2.1.Consistências locais (nó, arco, domínio/limites)

2.2.2.Consistências não-locais (caminho, k)

2.2.3.Algoritmos para a sua manutenção

2.3.Heurísticas para Pesquisa

2.3.1.Seleção de variável e valor

2.3.2.Heurísticas genéricas (dom, deg, dom/wdeg, impact)

2.3.3.Otimização - Branch & Bound

3.Resolução Incompleta de Restrições - Pesquisa Local Restringida

3.1.Modelação declarativa de restrições

3.1.1.Invariantes e objetos diferenciáveis

3.1.2.Grau de satisfação de restrições

3.2. Heurísticas

3.2.1.Baseadas em variáveis ou em restrições

3.2.2.Ávidas e estocásticas

3.3.Meta-heurísticas

3.3.1.Recomeços

3.3.2.Pesquisa tabu

3.3.3.Arrefecimento simulado

3.3.4.Pesquisa com vizinhança variável

3.3.5.Colónias de formigas

3.3.5. Syllabus:

1.Decision and Optimization Problems.

1.1.Examples.

1.2.Complexity.

2.Complete Constraint Solving - Backtrack search

2.1.Types of constraints

2.1.1.Arithmetic

2.1.2.Disjunctive

2.1.3.Reified

2.1.4.Global

2.2.Constraint networks

2.2.1.Local consistencies (node, arc, domain/bounds)

2.2.2.Non-local consistencies (Path, k)

2.2.3.Algorithms to maintain them.

2.3.Search heuristics

2.3.1.Variable and value Selection

2.3.2.General purpose heuristics (dom, deg, dom/wdeg, impact)

2.3.3.Optimisation - branch & bound

3.Incomplete Constraint Solving - Constrained-Based Local Search

3.1.Declarative constraint modelling

3.1.1.Invariants and differentiable objects.

3.1.2.Degree of satisfaction of constraints

3.2.Heuristics

3.2.1.Variable-based vs. constraint-based

3.2.2.Greedy and stochastic

3.3.Meta-heuristics

3.3.1.Restarts

3.3.2.Taboo search

3.3.3.Simulated annealing,

3.3.4.Variable neighbourhood search.

3.3.5.Ant colony optimisation

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Nesta unidade curricular aprofundam-se os conhecimentos de pesquisa referidos como opcionais no ACM 2008 CS Curriculum na área de Sistemas Inteligentes (IS/AdvancedSearch [elective]), e que complementam os introduzidos em unidades anteriores.

A resolução de problemas complexos de decisão e de otimização tem sido muito melhorada pelos avanços recentes na área de programação por restrições, quer através de algoritmos completos com retrocesso, mais focados na satisfação de restrições e importantes em áreas de verificação, quer através de algoritmos incompletos, mais utilizados em otimização.

A matéria lecionada insere-se claramente nos objetivos de dar a conhecer aos estudantes os avanços recentes mas já consolidados nesta área, que lhes permitam identificar oportunidades de aplicação destas técnicas em variadas situações práticas, modelar os problemas identificados e resolvê-los de forma eficiente.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course addresses in more depth the topics of search referred to as elective in the ACM 2008 CS Curriculum in the area of Intelligent Systems (IS/AdvancedSearch [elective]), and that complement those introduced in previous courses.

Solving complex decision and optimisation problems have undergone significant improvements in recent years, taking advantage of corresponding advances in constraint programming, both with complete backtrack-based algorithms, more focussed in constraint satisfaction arising in many verification tasks, as well as with incomplete constrained-based algorithms, more typically applied in optimisation tasks.

The syllabus clearly fulfil the objectives of the course, which is meant to provide the students with the advanced, if consolidated, knowledge in this area, allowing them to detect opportunities to apply it in many situations, modelling the identified problems, and efficiently solve them.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas e práticas. Nas primeiras são lecionados os conceitos e técnicas relevantes bem como a forma como estão implementados na linguagem e sistema de modelação e resolução utilizada (COMET).

Nas aulas práticas são resolvidos problemas retirados de uma biblioteca especializada (CSPLIB), sendo explorados modelos e técnicas de pesquisa alternativas disponibilizadas em COMET.

A avaliação de conhecimentos inclui:

- 2 testes individuais teóricos, onde é avaliado o conhecimento que os alunos adquiriram dos conceitos e características das técnicas lecionadas.*
- 2 trabalhos práticos que incidem sobre a resolução de problemas de restrições e otimização não triviais, e avaliam a capacidade dos alunos em utilizar e adaptar algoritmos completos (com retrocesso) e incompletos (pesquisa local restringida), explorados nas aulas práticas em problemas mais simples.*

3.3.7. Teaching methodologies (including evaluation):

The syllabus is taught in theoretical and laboratory classes. In the former, the main concepts and techniques are addressed, together with their implementation in the modelling and execution language and system adopted

(COMET).

The laboratory classes are dedicated to solving problems, selected from a specialised benchmark library (CSPLIB), using alternative models and search algorithms provided by COMET.

The evaluation of students include:

- *2 mid/final term tests, where the knowledge acquired by the students on the concepts and characteristics of the search techniques are assessed.*
- *2 practical projects, that focus on the resolution of non-trivial constraint and optimisation problems, and assess the ability of the students to use and adapt complete backtracking algorithms and incomplete constrained-based local search algorithms, explored in the lab classes on simpler problems.*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular identificam a) a aquisição de conhecimento recentemente desenvolvido para a satisfação de problemas de decisão e de otimização sujeitos a restrições; b) a experimentação da resolução de problemas, com base em modelos e algoritmos de pesquisa alternativos; e c) avaliação da adequação e eficiência das soluções alternativas, identificando-se as mais apropriadas para diferentes situações.

O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição dos vários conceitos e técnicas de programação de restrições e de pesquisa local baseada em restrições são lecionados, sendo estudadas as suas principais propriedades (por exemplo, complexidade). Estas matérias são avaliadas em testes individuais, onde os alunos têm um tempo limitado para mostrar o seu grau de assimilação das matérias lecionadas.

O comportamento de diferentes modelos e técnicas de pesquisa a aplicações concretas, requer uma aprendizagem mais baseada na experimentação, já que resultados teóricos existentes (por exemplo, o estudo da complexidade no pior caso) são pouco informativos. Para esse efeito, essa experimentação é feita nas aulas práticas em problemas razoavelmente simples, sendo avaliada através de trabalhos em aplicações não triviais, e que obrigam os estudantes a desenvolver capacidades de análise e de decisão bastante importantes nesta área.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The objectives of this course identify a) the acquisition of recently developed knowledge to solve constraint-based decision and optimisation; b) the experience obtained in solving such problems, adopting alternative models and search techniques; and c) assessment of the adequacy and efficiency of alternative approaches, with the identification of the most appropriate for different situations.

Objective a) is basically achieved in the theoretical classes, where the main concepts and techniques from both constraint programming and constrained based local search are explained, together with the study of their main characteristics (e.g. complexity). These topics are assessed in 2 individual mid/final-term tests, where the students have a limited time to show their level of assimilation of the relevant knowledge.

Such time limitation is inadequate to study the application of different models and techniques, which requires a more experiment-oriented training, since most theoretical results available (e.g. worst case complexity) have limited application. For this purpose, practical experimentation with simple problems is done in the laboratory classes, and is assessed in more complex projects, requiring the students to develop analysis and decision skills of great relevance in this area.

3.3.9. Bibliografia principal:

Livros de Texto / Text Books

- *Dina Rechter, Constraint Processing, Morgan Kaufman, 2003.*
- *Constraint-Based Local Search, MIT Press, 2005.*

Manuais /Manuals

- *Comet Tutorial, Dynamic Decision Technologies Inc., August 28, 2009*

Mapa IV - Programação Multiparadigma / Multiparadigm Programming

3.3.1. Unidade curricular:

Programação Multiparadigma / Multiparadigm Programming

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:
Artur Miguel de Andrade Vieira Dias (Regente) (T-28h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

O desenvolvimento de software sofisticado recomenda o domínio dum conjunto de paradigmas e técnicas de programação, idealmente suportados a nível linguístico de forma unificada numa linguagem multiparadigma.

Saber:

- 1 - Compreender as características, motivações e oportunidades de aplicação dos paradigmas de programação dominantes.*
- 2 - Compreender as vantagens de se ser fluente em diversos estilos de programação e em usá-los de forma integrada.*
- 3 - Compreender os princípios gerais que orientam o crescimento duma linguagem multiparadigma.*

Fazer:

- 4 - Conseguir desenvolver programas complexos, explorando bem os ingredientes linguísticos disponíveis.*
- 5 - Com sentido crítico, procurar soluções que se possam exprimir de forma simples e direta.*
- 6 - Capacidade de usar os paradigmas de forma isolada ou integrada.*
- 7 - Adquirir experiência no uso um sistema de programação multiparadigma particular: Scala.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

The development of sophisticated software recommends the mastery of a relatively large set of programming paradigms and techniques, if possible used in the unified framework of a multiparadigm language.

Knowledge:

- 1 - Understand the characteristics, motivations and opportunities of application of some of the dominant programming paradigms.*
- 2 - Understand the advantages of being fluent in diverse programming styles and use them in an integrated way.*
- 3 - Understand the general principles that govern the growth of a multiparadigm language.*

Application:

- 4 - Be able to develop complex programs, exploring available linguistic ingredients.*
- 5 - With a critical mind, be able to find solutions that can be expressed as simply and directly as possible.*
- 6 - Ability to use the paradigms alone or integrated.*
- 7 - Acquire experience in the use a particular multiparadigm programming system - Scala.*

3.3.5. Conteúdos programáticos:

A - Programação multiparadigma

Caracterização, motivação e exemplos.

Princípios orientadores na adição de novos conceitos.

B - Programação funcional

Estratégias de avaliação. Avaliação lazy e estruturas de dados infinitas.

Streams. Compreensões.

Tipos de dados algébricos e pattern-matching.

Continuações. Monadas.

C - Programação imperativa

Estado. Controlo. Iteradores.

D - Programação orientada pelos objetos

Classes. Traits. Mixins.

Polimorfismo familiar.

E - Tipos

Tipos nominais. Tipos estruturais. Tipos compostos com refinamentos.

Subtipos. Views.

Polimorfismo F-restringido. Limites inferiores. Anotações de variância.

F - Programação para a WEB (domínio específico)

Processamento de XML usando padrões baseados em expressões regulares.

G – Programação concorrente

Modelo dos Atores.

Futuros. Computações paralelas. Canais síncronos e assíncronos.

Paralelismo de dados.

3.3.5. Syllabus:

*A - Multiparadigm programming
Characterization, motivation and examples.
The Creative Extension Principle.*

*B - Functional Programming
Evaluation strategies. Lazy evaluation and infinite data structures.
Streams. Comprehensions.
Algebraic data types and pattern-matching.
Continuations. Monads.*

*C - Imperative programming
State. Control. Iterators.*

*D - Object-oriented programming
Classes. Traits. Mixins.
Family polymorphism.*

*E - Types
Nominal types. Structural types. Compounds types with refinements.
Subtypes. Views.*

F-bounded polymorphism. Lower bounds. Variance annotations.

*F - WEB Programming
XML processing using patterns based on regular expressions.*

*G – Concurrent programming
The Actor model.
Futures. Parallel computations. Asynchronous and Synchronous Channels.
Data parallelism.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A disciplina começa com uma discussão sobre a motivação para inventar novos conceitos de programação e sobre quais são os princípios a respeitar no crescimento duma linguagem. O tópico A do programa da disciplina suporta diretamente o objetivo 3 e oferece orientações para os objetivos 1, 2 e 6.

Depois, a maior parte do tempo da disciplina é usado para estudar, analisar e treinar a utilização de alguns paradigmas isoladamente e de forma combinada. Para cada paradigma, estudam-se os conceitos e técnicas clássicos (e.g. compreensões e continuações), mais alguns desenvolvimentos recentes (e.g. traits e polimorfismo familiar, ambos num contexto de tipificação estática). Os tópicos, de B a G do programa, situam-se todos ao mesmo nível e globalmente suportam os objetivos da disciplina.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The course begins with a discussion on the motivation to invent new programming concepts and what are the principles to be observed in the growth of a language. The topic A of the syllabus directly supports the objective 3 of the course and provides guidance for the objectives 1, 2 and 6.

Then, most of the time the course is used to study and train the use of a number of paradigms in isolation and in combination. For each paradigm, the study focuses on classical concepts and techniques (e.g. comprehensions and continuations), plus some recent developments (e.g. traits and family polymorphism, both in the context of static typing). The topics, from B to G of the syllabus, are all at the same level offer overall support to the goals of the course.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas, os conceitos fundamentais da cadeira são transmitidos, exemplificados e discutidos. Nas aulas práticas, os alunos resolvem pequenos problemas onde aplicam os conceitos e técnicas estudados. As aulas teóricas são dadas em sala de aula equipada com um projetor. As aulas práticas são realizadas em salas equipadas com computadores e projetor.

Os trabalhos da cadeira são realizados principalmente fora das aulas.

A avaliação consiste em dois testes (cada um com peso de 27.5%) e em três trabalhos de grupo de 2 alunos (trabalhos de dimensão pequena/média, cada um com peso de 15%). A nota de frequência é a média das notas dos três trabalhos. Para o aluno obter aprovação, tem de ter nota positiva de frequência, assim como nota positiva na média dos dois testes. Se o aluno for a exame de recurso (para obter aprovação ou melhorar a nota), este exame substitui os dois testes nas contas anteriores.

3.3.7. Teaching methodologies (including evaluation):

In the lectures, the concepts are presented, exemplified and discussed. In the lab classes, the students solve

small problems, applying the concepts and techniques learned. Lectures are given in a class room equipped with a projetor. In the lab rooms there are computes for the students and a projetor for the teacher.

The projects are mainly developed outside the classes.

The assessment consists of two tests (each weighing 27.5%) and three projects for teams of two students (small/average sized projects, each weighing 15%). The attendance grade is the average of the grade of the three projects. To pass, the student must have a minimal attendance grade of 9.5, as well as a minimum of 9.5 on the average grade of the two tests. If the student takes a ressit exam (either for passing or to improve the grade), this exam replaces the two tests in the previous rules.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas teóricas os alunos são iniciados nos conceitos e técnicas da cadeira. A interação entre o professor e os alunos promove a desejada construção do conhecimento e o desenvolvimento de bases críticas para autonomia de pensamento.

A solidificação dos conhecimentos, aptidões e competências ocorre de forma mais essencial nas aulas práticas e na execução dos trabalhos da cadeira. Os exercícios e trabalhos cobrem praticamente toda a matéria e incluem desafios que conduzem os alunos a compreender melhor os conceitos e a usá-los de forma apropriada.

Esta disciplina aproveita o facto da generalidade dos alunos já ter alguma experiência e maturidade, algo que interessa desenvolver ainda mais.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In the lectures, the students are initiated into the concepts and techniques of the course. The interaction between teacher and students promote the desired knowledge construction and the development of criticism bases for the autonomy of thought.

The consolidation of knowledge, abilities and skills occurs most essentially in the lab classes and during the work on the projects. The exercises and the projects cover almost the entire contents of the course; some of these exercises and projects include challenges that lead students to a better understand of the concepts and how to use them appropriately.

This course benefits from the fact that the majority of students have already some experience and maturity, something that should be further developed.

3.3.9. Bibliografia principal:

Text Book References:

- *Martin Odersky, Lex Spoon, and Bill Venners, "Programming in Scala", 2nd edition, Artima, 2010.*
- *Martin Odersky, "Scala by Example", Programming Methods Laboratory EPFL, Switzerland, 2011.*
- *Martin Odersky, "The Scala Language Specification", version 2.9, Programming Methods Laboratory EPFL, 2011*
- *Peter van Roy, Seif Haridi, "Concepts, Techniques, and Models of Computer Programming", The MIP Press, Additional Materials:*
- *Course notes, including exercises.*

Mapa IV - Qualidade de Software / Software Quality

3.3.1. Unidade curricular:

Qualidade de Software / Software Quality

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Ana Maria Diniz Moreira (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Miguel Carlos Pacheco Afonso Goulão (Regente) (T-28h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Componentes de um sistema de gestão da qualidade (SGQ)*
- *Técnicas de apoio à garantia da qualidade e sua automatização*

- *Metodologias de avaliação e melhoria do processo*
- *Modelos de qualidade do produto e avaliação das suas características*
- *Características de ferramentas de suporte à garantia da qualidade*

Saber fazer:

- *Reificar um SGQ para projetos de desenvolvimento de software*
- *Selecionar ferramentas de suporte à garantia da qualidade*
- *Conduzir uma avaliação da maturidade de um processo concreto*
- *Construir um modelo de qualidade para um produto específico proposto*
- *Selecionar normas aplicáveis a projetos de desenvolvimento*

Competências Complementares:

- *Coordenação intergrupala*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Components of a quality management system (QMS)*
- *Techniques to support quality assurance and their automation*
- *Process assessment and improvement methodologies*
- *Product quality models and assessment of their characteristics*
- *Characteristics of tools to support quality assurance*

Application:

- *Reify a QMS for software development projects*
- *Select tools to support quality assurance*
- *Conduct a maturity assessment of a concrete process*
- *Construct a quality model for a specific product*
- *Select applicable standards to software development projects*

Soft-skills:

- *Intergroup coordination*

3.3.5. Conteúdos programáticos:

1. *Qualidade de software: definição e objetivos*
2. *Planeamento, garantia e controlo de qualidade*
3. *Modelos de qualidade de software*
4. *Gestão da qualidade e certificação de processos*
5. *Modelos de maturidade e melhoria do processo (e.g. PSP, TSP, CMMI)*
6. *Avaliação qualitativa e quantitativa do processo de desenvolvimento de software*
7. *Avaliação qualitativa e quantitativa do software (produto)*
8. *Gestão da qualidade e certificação de produtos*
9. *Conceitos de verificação e validação*
10. *Inspeções, revisões e auditorias*
11. *Testes de Software*
12. *Reutilização de software e seu impacto na qualidade*
13. *Qualidade na evolução de software*
14. *Fiabilidade de Software*

3.3.5. Syllabus:

1. *Software Quality: definition and objectives*
2. *Quality planning, assurance, and control*
3. *Software Quality Models*
4. *Quality management, and processes certification*
5. *Software maturity and improvement models (e.g. PSP, TSP, CMMI)*
6. *Qualitative and Quantitative software development process evaluation*
7. *Qualitative and Quantitative software (as a product) evaluation*
8. *Quality management, and product certification*
9. *Verification and validation concepts*
10. *Inspections, reviews, and audits*
11. *Software testing*
12. *Software Reuse and its impact on quality*
13. *Quality in software evolution*

14. Software reliability

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Apresenta-se uma visão pragmática da qualidade, em linha com as boas práticas da indústria. O programa inclui uma introdução à qualidade de software (1-3), explorada quer numa vertente mais dirigida ao processo de desenvolvimento (4-7), quer numa focada no produto (8-14). Estudam-se os principais elementos de um sistema de gestão de qualidade (4,8). Discutem-se modelos de maturidade do processo e abordagens de avaliação e melhoria do mesmo (5, 6). A implementação de tais modelos envolve o desenvolvimento de práticas que visam melhorar a qualidade do produto. São discutidas diversas técnicas de verificação e validação (9-11). A avaliação da qualidade do software enquanto produto incide sobre aspetos fundamentais tais como a sua facilidade de manutenção e reutilização, e a sua fiabilidade (8, 12-14). O estudo da qualidade é suportado por normas, metodologias e modelos de qualidade de grande impacto nas organizações.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

A pragmatic view of quality, in line with industry's best practices is presented. The syllabus includes an introduction to software quality (1-3), explored both through a process (4-7) and a product (8-14) perspective. The most important elements of a software quality management system are studied (4, 8). Software maturity models, as well as how to evaluate and improve the process are explored (5, 6). The implementation of those maturity models involves using techniques aimed at fostering product quality. Verification and validation techniques are discussed (9-11). Software product quality evaluation is focused in fundamental elements such as its maintainability, reuse and reliability (8, 12-14). The perspective on software quality in this course is anchored by standards, methodologies and quality models with a deep impact in organizations.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas apresenta-se a matéria, complementando os conceitos apresentados com exemplos concretos recolhidos de organizações de desenvolvimento de software em larga escala. Nas aulas práticas é dado ênfase à utilização de técnicas de promoção da qualidade, bem como a técnicas de avaliação da qualidade do processo e do produto. Estas técnicas são introduzidas e enquadradas previamente nas aulas teóricas, naturalmente. O processo de avaliação conta com dois trabalhos práticos (valendo 20% da nota final cada um) e dois testes (valendo 30% da nota final cada um). Os trabalhos são realizados em grupo e visam a aplicação prática de algumas das técnicas de avaliação e promoção da qualidade do produto e do processo. Para obter aprovação, os alunos devem ter média igual ou superior a 9,5 valores nos trabalhos e média igual ou superior a 9,5 valores nos testes.

3.3.7. Teaching methodologies (including evaluation):

The course subjects are presented in the lectures, where the theoretical concepts are illustrated by concrete examples taken from large scale software development organizations. The lab sessions focus in the usage of concrete techniques for promoting and evaluating software process and product quality. These techniques are previously introduced and contextualized in the lectures. The evaluation process includes two projects, each worth 20%, as well as two tests, each worth 30%. The projects are carried out in groups and involve the application in practice of some of the studied software quality evaluation and promotion techniques. To be approved, students must obtain an average grade greater or equal to 9,5 points in the projects and an average grade greater or equal to 9,5 points in the tests.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A apresentação das matérias nas teóricas é complementada com a exercitação de técnicas particulares que as implementam nas aulas práticas (e nos projetos). Deste modo, os alunos são expostos a boas práticas usadas nas organizações de desenvolvimento de software com um elevado grau de maturidade e têm a oportunidade de experimentar, em concreto, algumas dessas práticas. Os trabalhos práticos a desenvolver em grupo visam consolidar a utilização dessas práticas, contribuindo ainda para o desenvolvimento de competências relacionadas com o trabalho em equipa e em colaboração entre diferentes equipas, em particular, no contexto de atividades de verificação e validação. Este tipo de colaboração entre equipas contribui para fortalecer aspetos tais como a coordenação intergrupala e a defesa pública de pontos de vista, sendo ambos os aspetos instrumentais nas atividades de promoção e garantia da qualidade de software. Por outro lado, a consolidação de conceitos mais teóricos que suportam as boas práticas na indústria é aferida sobretudo através dos testes de avaliação.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The introduction of subjects in the lectures is complemented with the usage of particular techniques that implement those subjects in the lab sessions (and in the projects). This way, students are exposed to best practices from software development organizations with a high maturity level, and are able to experiment some of those techniques.

The projects are carried out in groups and in cooperation with different groups, particularly in the context of verification and validation tasks.

This kind of cooperation is aimed at interteam cooperation and the public defense of points of view, both aspects being instrumental in software quality promotion and assurance. Furthermore, the consolidation of the more theoretical aspects that support these best practices in industry is assessed mostly through the evaluation tests.

3.3.9. Bibliografia principal:

1. Linda Westfall, "The Certified Software Quality Engineer Handbook", Quality Press, 2009
2. Stephen K. Han, "Metrics and Models in Software Quality Engineering", Addison-Wesley Professional, 2nd Edition, 2002.

Mapa IV - Redes de Computadores TCP/IP / TCP/IP Networking

3.3.1. Unidade curricular:

Redes de Computadores TCP/IP / TCP/IP Networking

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Augusto Legatheaux Martins (T-28h; P-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

N/A

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Esta unidade curricular é uma segunda disciplina em redes de computadores para estudantes do 3º ou 4º ano do mestrado integrado, com o objetivo de permitir uma compreensão mais aprofundada do tema. O curso também comporta uma vertente aplicada e experimental com uma atividade laboratorial extensa. Projetos de grupo permitem aos estudantes ganhar um primeiro contacto com tópicos emergentes e avançados. Os estudantes deverão já ter seguido com sucesso uma primeira unidade curricular de introdução às redes de computadores.

Saber:

- *Transmissão e comutação de pacotes de dados*
- *Encaminhamento*
- *Qualidade de serviço e dimensionamento da rede*
- *Protocolos de transporte*
- *Tópicos emergentes em redes de computadores*

Saber Fazer:

- *Capacidade de montar uma pequena rede TCP/IP*
- *Utilização de ferramentas de análise da rede e medida do seu desempenho*
- *Contacto com equipamento representativo*
- *Capacidade de analisar e investigar tópicos emergentes*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This is a second course on computer networks, for advanced under-graduated or graduated students, allowing students to gain a deeper comprehension of this subject. The course also takes a hands-on approach with extensive laboratory and project activity. Group projects allow students to gain insights in hot and emergent computer networking topics. Preferably, students should have previously succeeded an introductory level course on Computer Networks.

Knowledge:

- *Transmission and packet switching*
- *Routing*
- *Quality of service and network dimensioning*
- *Transport protocols*
- *Emergent topics in computer networking*

Application:

- Ability to setup a small TCP/IP Network
- Ability to use tools for network analysis and measurement
- Awareness of representative equipment used in real computer networks
- Research and analyze emergent networking topics

3.3.5. Conteúdos programáticos:

1. *Introdução. Características essenciais das redes TCP/IP*
2. *Serviços do nível canal. Protocolos de partilha dos meios de transmissão. Switched Ethernet, SPT e VLANs. Soluções avançadas para rede Ethernet de grande dimensão.*
3. *Topologia e encaminhamento em redes TCP/IP. Topologia. Protocolos de encaminhamento Intradomain, interdomain and multicasting. Aspetos avançados: traffic engineering, edge controlled and overlay routing*
4. *Desempenho e qualidade de serviço. Multimedia streaming e voice-over-IP. Qualidade de serviço ao nível rede e de transporte.*
5. *Protocolos de transporte. Revisões e evolução recente do controlo da saturação em TCP. Novos protocolos de transporte.*

As sessões laboratoriais abrangem tópicos desde o nível canal até ao estudo do desempenho e da qualidade de serviço e têm lugar num laboratório especialmente adequado a esse fim. Tópicos emergentes e de relevância são também propostos como temas de projeto de grupo.

3.3.5. Syllabus:

1. *Introduction. Fundamentals of TCP/IP networks*
2. *Link layer services. Shared media protocols. Switched Ethernet, SPT and VLANs. Advanced solutions for large scale switched networks*
3. *Topology and Routing in IP Networks. Topology. Intradomain, interdomain and multicasting routing protocols. Advanced issues: traffic engineering, edge controlled and overlay routing*
4. *Performance and Quality of Service. Multimedia streaming and voice-over-IP. Quality of Service at the network and transport levels.*
5. *Transport Protocols. Review and evolution of congestion control in TCP. New transport protocols. Laboratory sessions on the topics from the link-layer up to performance and quality of service take place in a special purpose laboratory. Emergent and hot topics will be proposed as subjects for group projects.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O curso tem por objetivo permitir a aprendizagem aprofundada pelo aluno dos temas clássicos de redes de computadores dado que se trata de uma unidade curricular opcional para alunos do mestrado integrado (a unidade curricular recebe de 20 a 40 estudantes em cada ano). Adicionalmente, dois outros objetivos são atingidos. Por um lado permitir aos alunos um contacto com switches, routers e as ferramentas usadas para teste e validação do funcionamento da rede. Este objetivo é atingido através das sessões laboratoriais. O segundo objetivo é levar os estudantes a contactarem com tópicos emergentes e relevantes. Este segundo objetivo é atingido através da realização de um projeto de grupo que comporta trabalho de investigação e de implementação.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course aims to expose students to the classical networking subjects to allow their deeper understanding since students take the course as an elective at senior level (the course receives from 20 to 40 students each year). Additionally, two other goals are to be fulfilled. On one hand a first contact with switches and routers as well as the tools used to test and validate network functioning and performance. This goal is mainly accomplished with the lab sessions. On the other hand students are exposed to new and emergent topics. This goal is accomplished with the realization of a group project that can encompass research as well as implementation work.

3.3.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas têm por objetivo a apresentação dos principais temas e discutir as questões mais relevantes. Os estudantes devem, de preferência antes das aulas, estudar a bibliografia recomendada.

As sessões laboratoriais têm lugar num laboratório especializado e têm por objetivo montar diversas configurações de rede e testar o seu desempenho. O laboratório está equipado com equipamento atual, idêntico ao que é usado na indústria para montar redes de pequena e média dimensão. Estas sessões laboratoriais têm uma aproximação do tipo aprendizagem através de "mão na massa".

Os projetos de grupo permitem aos estudantes ganhar competências em temas emergentes em redes de computadores.

Componentes da avaliação:

- *Dois testes intermédios (com o peso total de 50% na nota final)*
- *Trabalhos laboratoriais (com o peso total de 25% na nota final)*
- *Projeto de grupo (com o peso total de 25% na nota final)*

3.3.7. Teaching methodologies (including evaluation):

Lectures are intended to support the instructor's presentation of fundamental issues. Students should, preferably before classes, study the recommended bibliography.

Laboratory sessions take place in a specialized laboratory and their aim is to perform several network configurations and test their performance. The laboratory is equipped with state of the art network gear, identical to the one currently used in industry in small to medium networks. These lab sessions are based on a "hands-on" computer network learning approach.

Group projects allow students to gain insights in hot and emergent computer networking topics.

Assessment Components:

- *Two intermediate tests (50 % of the final grade)*
- *Written and lab assignments (25% of the final grade)*
- *Group research (and implementation) project (25% of final grade)*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Cada um dos objetivos de aprendizagem é tratado com especial cuidado. A compreensão e o estudo dos tópicos principais é objeto das aulas teóricas e avaliado através de testes. Os estudantes devem atingir o domínio dos conceitos através do estudo dos livros e dos artigos científicos clássicos fornecidos. As sessões laboratoriais confrontam os estudantes com a implementação real, parametrização dos equipamentos e o teste dos protocolos. Durante essas sessões diversas configurações de rede são atestadas e analisadas e o seu funcionamento e desempenho é avaliado. Finalmente, tópicos mais avançados e emergentes são alvo de investigação e testados através de projetos de grupo.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Each of the learning outcomes is carefully treated. Understanding and comprehension of the main topics is dealt by lectures and evaluated with tests. Students must attain a reasonable level of expertise by individual study of textbooks and classical papers. Laboratory sessions allow students to be confronted with a real world implementation and field test of protocols and networking equipment configuration. During these sessions several network configurations are setup, tested and analysed in what concerns their functioning and performance. Finally, more advanced topics and new and emergent proposals are researched (and tested) by the way of a group project.

3.3.9. Bibliografia principal:**Course support book**

- *Larry L. Peterson and Bruce S. Davie, "Computer Networks – A Systems Approach – 4th Edition," Morgan & Kaufman, 2007*

Other books

- *James F. Kurose and Keith W. Ross, "Computer Networking - A Top-Down Approach Featuring the Internet," Addison Wesley Pearson, Inc., 5th Edition, 2010*

The instructor also provides several classical papers to be studied during the course or to support projects.

Mapa IV - Requisitos e Arquitetura de Software / Software Requirements and Architecture**3.3.1. Unidade curricular:**

Requisitos e Arquitetura de Software / Software Requirements and Architecture

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Ana Maria Dinis Moreira (T-14h; P-14h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Baptista da Silva Araújo Júnior (T-14h; P-14h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Objetivo: estudo dos vários tipos de requisitos como pilares orientadores da arquitetura de software. Os requisitos requisitos não-funcionais (NFRs) e análise de conflitos servem de base para derivação sistemática da arquitetura de software.

Saber:

- Lidar com desafios colocados pela dimensão, objetivos contraditórios e descrição deficiente dos problemas reais.
- Compreender as vantagens da sistematização e reutilização planeada para a derivação da arquitetura.
- Ponderar a arquitetura com base em alternativas nem sempre disjuntas.

Saber Fazer:

- Aplicar abordagens orientadas a objetivos para identificar, especificar, validar e gerir requisitos.
- Usar o NFR Framework para refinar NFRs e resolver conflitos.
- Analisar alternativas arquiteturais e escolher para satisfazer as restrições impostas pelos NFRs, promovendo a derivação sistemática da arquitetura.
- Descrever arquiteturas usando visões, padrões e estilos.
- Utilizar linguagens de descrição arquitetural ferramentas automáticas.

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Objective: study the various types of requirements as drivers of the software architecture. Non-functional requirements (NFRs) and conflict analysis form the basis for the systematic derivation of the software architecture.

Knowledge:

- Deal with the challenges imposed by the size, conflicting goals and poor requirements.
- Understand the advantages of systematic development and planned reuse.
- Consider the structure based on a range of possibly disjoint alternatives.

Application:

- Apply goal-oriented approaches to identify, specify and validate requirements.
- Use the NFR Framework to refine NFRs and resolve conflicts.
- Analyze the architectural choices and choose to meet the restrictions imposed by the NFRs, promoting the systematic derivation of the software architecture.
- Specify software architectures using views, patterns, styles.
- Use existing architectural description languages and automated tools.

3.3.5. Conteúdos programáticos:

1. Introdução aos tipos de requisitos

- A importância da Engenharia de Requisitos
- que é a Arquitetura de Software e sua importância para evolução
- Contextualização do paradigma orientado pelos objetivos

2. Técnicas de elicitação, análise e descrição orientadas a objetivos

- Validação de requisitos (inspeções, protótipos)
- Gestão de requisitos (requisitos voláteis e estáveis, gestão de mudança e rastreabilidade)
- Análise de NFRs (refinamento com NFR Framework e gestão de conflitos)

3. Arquitetura do software e conformidade com requisitos de qualidade

- Mapeamento de requisitos para arquitetura
- Visões Arquiteturais
- Estilos e padrões Arquiteturais (catálogos de arquiteturas)
- Linguagens de descrição arquitetural
- Desenho, avaliação e refinamento de arquiteturas

4. Reutilização de requisitos e arquitetura

- Análise de domínio
- Linhas de produto de software
- Frameworks
- Padrões de análise
- Modularização avançada
- Problem Frames
- Componentes

3.3.5. Syllabus:

1. Introduction to various types of requirements

- *The importance of Requirements Engineering*
- *What is Software Architecture and its importance in evolution*
- *Brief background of the goal-oriented paradigm*
- 2. *Techniques for requirements elicitation, analysis and specification*
 - *Validation of requirements (inspections, prototypes)*
 - *Requirements Management (volatile and enduring requirements, change management and traceability)*
 - *Analysis of NFRs (NFR Framework for refinement and conflict management)*
- 3. *Software architecture and compliance with quality requirements*
 - *Mapping requirements to architecture*
 - *Architectural views*
 - *Architectural styles and patterns (catalogs of software architectures)*
 - *Architectural description languages*
 - *Design, evaluation and refinement of architectures*
- 4. *Reuse of requirements and architecture*
 - *Domain Analysis*
 - *Software Product Lines*
 - *Frameworks*
 - *Analysis Patterns*
 - *Advanced Modularization*
 - *Problem Frames*
 - *Components*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O objetivo desta UC é o estudo dos vários tipos de requisitos como pilares orientadores da arquitetura de software no contexto do estado da arte atual, envolvendo por isso:

- *o estudo de técnicas próprias de engenharia de requisitos dirigida por objetivos com ênfase na especificação e análise de requisitos não-funcionais que são fundamentais para a derivação informada e sistemática da arquitetura do software (Partes 1 e 2);*
- *a análise e propagação dos resultados para a arquitetura de software com vista a facilitar a evolução do futuro sistema leva ao estudo das visões, padrões e estilos arquiteturais, terminando com o refinamento e avaliação da arquitetura resultante (Partes 1, 3 e 4); e*
- *culminando numa visão global do estado da arte atual (Parte 4).*

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The purpose of this UC is the study of the various types of requirements as drivers of the software architecture in the context of the current state of the art. This involves:

- *the study of goal-oriented requirements engineering techniques with emphasis on the specification and analysis non-functional requirements, what is fundamental to the informed and systematic derivation of the software architecture (Parts 1 and 2);*
- *the analysis and propagation of results to the software architecture to facilitate the development of future system leads to the study of the views, patterns and architectural styles, finishing with the refinement and evaluation of the resulting architecture (Parts 1, 3 and 4); finally,*
- *the whole course culminates with an overview of the current state of the art (Part 4).*

3.3.7. Metodologias de ensino (avaliação incluída):

No início das aulas os alunos recebem os tópicos de investigação e o trabalho prático. Um prazo rígido é dado para a entrega dos trabalhos.

As aulas teóricas são dadas em sala de aula equipada com um projetor. As aulas práticas são realizadas num laboratório equipado com computadores e projetor.

A avaliação consiste das seguintes partes obrigatórias: um trabalho prático; um relatório sobre um tópico de investigação; dois testes. O trabalho e o relatório são realizados em grupo e os testes individualmente.

A nota é uma média ponderada do trabalho prático (40%), do relatório de investigação (20%) e da nota dos testes (20% cada um). Os alunos que não cumprirem qualquer uma das partes acima são classificados como "Reprovados" e "Ausentes" caso não entreguem qualquer componente de avaliação.

Acesso a exame de recurso é reservado a alunos com desempenho satisfatório (com média final entre 9,5 e 20).

3.3.7. Teaching methodologies (including evaluation):

In the first week the students receive the schedule for their research topics and practical work. A hard deadline is defined for both cases.

Lectures are given in a room equipped with a computer projetor. Practical sessions take place in a lab equipped

with computers and projector.

Assessment consists of four mandatory parts: a practical work; an essay on a research topic; 2 tests. Practical works and essays are done in groups and the tests are individual.

The final grade is the weighted average of the grades obtained in the practical work (40%), the essay (20%) and the tests (20% each). Students who do not submit a component of the evaluation process are classified as "Reprovado" (fail), while others, that did not take part in any evaluation process are classified with "Ausente" (absent).

Access to the ressit exam is conditional upon the satisfactory performance with a grade of at least 9,5 out of 20.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As matérias apresentadas nas aulas teóricas são ilustrados nas práticas com exemplos inspirados em situações reais, garantindo a consolidação dos vários tópicos. O trabalho prático, de dimensão significativa, permite ao aluno experimentar, de forma independente e em concreto, as técnicas ensinadas desde a elicitação de requisitos até ao desenho da arquitetura e sua avaliação. Os trabalhos salientarão a importância dos requisitos não funcionais e seus tradeoffs, levando o aluno compreender a necessidade de usar estruturas standard (como padrões, estilos) para o desenho da arquitetura.

Neste contexto, os alunos compreenderão ainda as vantagens da reutilização planeada. O artigo investigação desafia a curiosidade e vontade do aluno em ir para além do que é ensinado nas aulas. Os milestones do trabalho prático e artigo serão verificados na aula prática. O trabalho prático e o relatório de investigação fomentam o trabalho em equipa, os testes avaliam o desempenho individual.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The topics presented in lectures are illustrated in the lab with examples inspired by real situations, ensuring the consolidation of the various topics. A project, of a significant size, provides an opportunity for students to experiment the techniques taught, from requirements elicitation to the design of software architecture and its evaluation.

The examples stress the importance of non-functional requirements and their required tradeoffs, allowing students to understand the need for standard architectural structures (such as patterns, styles). In this context, students should see the benefits of planned reuse. The research paper challenges students' curiosity and willingness to go beyond what is taught in class. Project and the paper milestones will be assessed in practical classes, so that the work progress is ensured. Both the project and the paper foster teamwork, while the tests assess individual performance.

3.3.9. Bibliografia principal:

- A.Lamsweerde, *Requirements Engineering*, Wiley, 2009
- I.Alexander, N. Maiden, *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*, Wiley, 2004
- G.Kotonya, I. Sommerville, "Requirements Engineering: Processes and Techniques", Wiley, 1998
- I.Alexander, R. Stevens, "Writing Better Requirements", Addison Wesley, 2002
- I.Sommerville, P. Sawyer, "Requirements Engineering: A Good Practice Guide", Wiley, 1997
- Proceedings of the IEEE Requirements Engineering Conference
- L.Bass, P. Clements, R. Kazman, "Software Architecture in Practice", 2nd edition, Addison-Wesley, 2003
- P.Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, J. Stafford, "Documenting Software Architectures: Views and Beyond", Addison-Wesley, 2003
- M.Shaw, D. Garlan, "Software Architecture: Perspectives on an Emerging Discipline", Prentice Hall, 1996
- M.Fowler, D. Rice, M. Foemmel, E. Heatt, R. Mee, R. Stafford, *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2002

Mapa IV - Reutilização e Evolução de Software / Software Reuse and Evolution

3.3.1. Unidade curricular:

Reutilização e Evolução de Software / Software Reuse and Evolution

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Ana Maria Diniz Moreira (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Miguel Jorge Tavares Pessoa Monteiro (Regente) (T-28h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Detetar e saber lidar com necessidade de evolução de sistemas*
- *Diferença entre reutilização oportunista e reutilização planeada*
- *Distinguir Re-Engenharia, Engenharia Reversa e Reutilização*
- *Técnicas de Análise de Domínio*
- *Industrialização do Desenvolvimento como Forma de Reutilização para economia de escala*

Saber Fazer:

- *Conhecer abordagens de desenvolvimento orientadas à reutilização (e.g., técnicas baseadas em componentes e técnicas orientadas a modelos)*
- *Usar técnicas e ferramentas de reengenharia em larga escala*
- *Modelação: desenvolver soluções efetivas para Re-Engenharia*
- *Analisar sistemas existentes e aplicar técnicas de Reutilização e Re-Engenharia*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge

- *Detect the need for software system's evolution and deal with it*
- *Understand the difference between opportunistic reuse and planned reuse*
- *Distinguish Reengineering, reverse engineering and reuse*
- *Learn Domain Analysis techniques*
- *Industrialization of the Development as a way to reuse in large-scale*

Application:

- *Reuse-oriented development approaches (e.g., component-based development, model-driven development)*
- *Techniques and tools for large-scale reengineering projects*
- *Modelling: develop effective solutions for Re-Engineering*
- *Analyse existing systems and apply Reuse and Re-Engineering techniques*

3.3.5. Conteúdos programáticos:

I – Introdução

- *Lidar com a mudança, complexidade do desenvolvimento em massa, e customização*
- *Reutilização Sistemática (oportunist vs. planeada)*
- *Desenvolvimento por composição de componentes reutilizáveis*
- *Reengenharia, Engenharia Reversa, Refabricação e Reutilização*

II - Análise de Domínio

III - Reutilização de Software

IV - Abordagens Avançadas de Reutilização

- *Famílias de Produtos de Software*
- *Desenvolvimento Orientado a Modelos*
- *Desenvolvimento Orientado a Aspectos e Modularidade Avançada*
- *Desenvolvimento de Software Orientado a Serviços*

V – Re-Engenharia de Software e Refabricação

VI - Estudos de casos em diferentes domínios de aplicação

3.3.5. Syllabus:

I – Introduction

- *Dealing with Change, Complexity, Mass development and customization*
- *Systematic Reuse (opportunistic vs. planned)*
- *Development by assembly*
- *Re-Engineering, Reverse Engineering, Refactoring and Reuse*

II - Domain Analysis

III - Software Reuse

IV – Advanced Reuse Approaches

- *Software Families*
- *Model-Driven Development*
- *Aspect-Oriented Software Development and Advanced Modularity*
- *Service Oriented Software Development*

V – Software Reengineering and Refactoring

VI - Case Studies in different application Domains

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O objetivo resumido desta unidade curricular é oferecer aos alunos meios que lhes permitam: (1) aprender técnicas e usar ferramentas de reutilização planeada em larga escala para o desenvolvimento massificado. (2) lidar com problemas relacionados com a evolução de sistemas, onde muitos dos exemplos serão casos de estudo originários da indústria.

As matérias estudadas nos tópicos II, III e IV oferecem os meios para possibilitar a reutilização planeada, sendo o desenvolvimento massificado apoiado grandemente pelo estudo das Famílias de Produtos. O tópico V cobre as técnicas reengenharia, incluindo a refabricação de software e técnicas de suporte associadas. Estas enquadram-se na evolução de sistemas existentes e legados. O tópico VI é pensado em garantir que os casos de estudo são variados e de origem industrial.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The purpose of this course is to provide students with the means to: (1) Learn to techniques for reuse on a large scale planned for the massive development, as well as associated tools. (2) Deal with problems relating to systems development, where many of the examples are meant to be case studies originating in industry.

Subjects studied topics in II, III and IV cover planned reuse, mass software development and supporting approaches such as Software Product Families. Topic V covers the technical reengineering and refactoring of existing and legacy systems, as well as the associated tool support. These are within the scope of the topic on software evolution. The topic VI is thought to ensure that the case studies are varied and of industrial origin.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas teóricas é exposta a matéria, ilustrada com exemplos de aplicação. As aulas de laboratório são usadas ferramentas em problemas, de preferência originados na indústria, e feitas diversas ilustrações e exercícios. O aluno terá de construir com sucesso um projeto de dimensão não-trivial em grupo. Este projeto será entregue (e avaliado) em duas partes diferentes. A nota final é calculada de acordo com a expressão:

Nota Final = Projeto - P1 (25%) + Projecto- P2 (25%) + 2 Testes (25% cada)

A condição para obter aprovação: $P \geq 9,5$ e $T \geq 9,5$, sendo P a média das duas partes do projeto e T a média dos dois testes.

As 4 componentes de avaliação são obrigatórias. O acesso ao exame de recurso requer que o aluno obtenha um

desempenho satisfatório com média entre 9,5 em 20. Melhorias de nota têm lugar apenas no exame de recurso, que nesses casos conta para 100% da nota final – na condição da nota obtida ser ≥ 9.5 .

3.3.7. Teaching methodologies (including evaluation):

Weekly lectures consist of a theoretical exposition of the topics, illustrated with the application of examples. Lab sessions (2 hours weekly) focus on resolution of problems, mostly originated in industry, and use of tools directly

connected to the topics discussed in the previous lectures.

Student must successfully build a project. This project will be developed in groups and delivered (and evaluated) in two different parts or phases. The project will include integration (reuse) of existing parts – which provide opportunities and scenarios for exercising reengineering and refactoring. Final grade (FG) is as follows:

$FG = \text{Project- } P_1 (25\%) + \text{Project- } P_2 (25\%) + 2 \text{ Testes } (25\% \text{ each})$

Condition for approval: $P \geq 9.5$ and $T \geq 9.5$, where P is the average of the 2 parts of the project and T the average of the 2 tests.

The 4 components of evaluation are required. Grade improvement is to be done in the appeal exam, worth 100% of the final grade – provided is ≥ 9.5 .

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As matérias estudadas nos tópicos II, III, IV e V são de natureza eminentemente prática, requerendo treino e tarefas com casos concretos – daí a existência de um projeto. Como os diversos tópicos podem contribuir para uma mesma tarefa, não há necessariamente um mapeamento um-para-um entre tópicos e as 2 fases do projeto. As 2 fases focam-se em partes distintas da matérias abordadas, mas pretende-se que a experiência ganha com a primeira fase do projeto também contribua, numa lógica de positive feedback loop, para a realização da segunda fase.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Materials for topics II, III, IV e V are of a primarily practical nature, thus requiring hands-on training with concrete systems. Hence the existence of a project. Since more than one topic can contribute to the same

task/assignment, there is no one-to-one mapping between topics and tasks. The two phases are meant to focus on different topics but it is also intended that the second phase will contribute, in a positive-feedback loop, to the tasks of the second phase of the project.

3.3.9. Bibliografia principal:

Text Book References:

- *Software Product Lines, Paul Clements, Addison-Wesley Professional; 3rd edition, August 30, 2001*
- *Software Evolution, Tom Mens, Serge Demayer, Springer, 1st ed. 2008 edition, November 19, 2010*
- *Software Language Engineering: Creating Domain-Specific Languages Using Metamodel, A.Kleppe, Addison-Wesley, 1st edition, 2008*
- *Software Reuse: Architecture, Process and Organization for Business Success by Ivar Jacobson, M. Griss and P. Jonsson, Jun 1, 1997*
- *Domain-Specific Modeling, Steven Kelly, J. Tolvanen, IEEE Wiley, 2008*
- *Generative Programming Methods, Tools and Applications, K. Czarnecki and U. Eisenecker, Addison-Wesley, 2000*
- *Software Factories, J. Greenfield and K. Short, Wiley Publishing, 2004*

Additional References:

- *Slides for the course.*
- *Set of proposed exercises*

Mapa IV - Segurança de Sistemas e Redes de Computadores / Computer Systems and Network Security

3.3.1. Unidade curricular:

Segurança de Sistemas e Redes de Computadores / Computer Systems and Network Security

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Rodrigo Seromenho Miragaia Rodrigues (T-14h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Henrique João Lopes Domingos (T-14h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Frameworks de referencia para serviços de segurança de sistemas e redes de computadores.*
- *Tópicos avançados de criptografia computacional aplicada.*
- *Serviços e protocolos de segurança em redes de computadores e sistemas distribuídos.*
- *Normas e serviços de segurança na pilha TCP/IP.*
- *Normas e serviços de segurança para redes sem fios.*
- *Tópicos emergentes e de investigação em soluções de segurança e confiabilidade para sistemas distribuídos.*

Saber Fazer:

- *Técnicas de programação com algoritmos e processos da criptografia computacional.*
- *Implementação de algoritmos e processos criptográficos.*
- *Conceção, implementação e análise experimental de protocolos e serviços de segurança para sistemas e redes.*
- *Ferramentas de teste e auditoria de conformidade de critérios de segurança.*
- *Avaliação experimental de soluções emergentes de segurança e confiabilidade para sistemas distribuídos de grande escala.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Security frameworks and standards for computer systems and networks security services;*
- *Advanced topics on applied computational cryptography.*
- *Implementation of cryptographic algorithms and processes.*
- *Network security services and protocols.*
- *TCP/IP security services and standards.*

- *Wireless Networks Security services and standards.*
 - *Emergent and research topics on dependability solutions for computer networks and distributed systems.*
- Application:*
- *Advanced programming practice with computational cryptography;*
 - *Design, implementation and experimental analysis of security protocols and services for computer systems and networks.*
 - *Practice with security components and security auditing and testing tools.*
 - *Experimental analysis of dependability solutions for large scale distributed systems.*

3.3.5. Conteúdos programáticos:

- 1.Arquiteturas e frameworks de segurança para sistemas e redes de computadores*
 - *Criptografia computacional aplicada*
 - *Algoritmos e métodos da criptografia computacional aplicada*
 - *Técnicas de implementação de métodos e protocolos criptográficos*
- 2.Protocolos e aplicações*
 - *Autenticação e certificação digital*
 - *Serviços de segurança na pilha TCP/IP: pilhas TLS, SSH e IPSec*
 - *Aplicações de segurança na pilha TCP/IP*
 - *Protocolos e aplicações de segurança para a Internet: transações eletrónicas seguras, sistemas de pagamentos e dinheiro digital*
 - *Serviços e normas de segurança para redes sem fios 802.11*
- 3.Segurança de sistemas*
 - *Sistemas de autenticação e SSO*
 - *Segurança e auditoria de software*
 - *TPMs e aplicações*
 - *Controlo de acessos*
 - *Sistemas firewall e sistemas confiáveis*
 - *VPNs e VLANs seguras*
 - *Soluções de segurança em centros de dados seguros*
 - *Avaliação de critérios segurança de sistemas*
 - *Tolerância com recuperação pró-ativa a intrusões*
 - *Soluções de confiabilidade para sistemas distribuídos*

3.3.5. Syllabus:

- 1.Security architectures, models, frameworks and standards*
 - *Applied computational cryptography*
 - *Advanced methods and algorithms*
 - *Cryptographic protocols and implementation techniques*
- 2.Network security protocols and applications*
 - *Authentication systems and digital certification*
 - *TCP/IP security stack and cross-layering security: SSL, SSH and IPSec suites*
 - *TCP/IP security applications*
 - *Internet security solutions: secure electronic transactions, payment systems and digital cash protocols*
 - *Security services and standards for 802.11 networks*
- 3.Systems security*
 - *Authentication and Single Sign On solutions*
 - *Software security and auditing*
 - *TPMs and applications*
 - *Access control*
 - *Perimeter defense systems and firewalls*
 - *DoS protection*
 - *Trusted systems*
 - *Secure VPNs e secure VLANs*
 - *Security solutions for secure Data-Centers*
 - *Systems common security criteria and evaluation*
 - *Intrusion tolerance and proactive recovering techniques*
 - *Dependability solutions for secure and reliable distributed systems*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Esta unidade curricular parte de conhecimentos introdutórios e valências iniciais associadas à problemática da segurança em sistemas computacionais e destina-se à aquisição e aprofundamento de conhecimentos avançados de segurança e confiabilidade em sistemas e redes de computadores e sistemas distribuídos.

A UC propicia uma aquisição em profundidade de tópicos avançados teóricos e práticos, dirigidos a valências de conceção, desenvolvimento, avaliação experimental e gestão de sistemas confiáveis e seguros, para suporte de sistemas distribuídos e aplicações confiáveis, operando em ambiente internet ou em infraestruturas de computação e comunicação de grande escala.

A aquisição de técnicas de conceção, desenvolvimento e avaliação de critérios de segurança com integração na pilha TCP/IP é particularmente focada e treinada em laboratório, com base em exercícios de programação, trabalhos e projetos de avaliação.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This is an advanced course on computer systems and network security, for under-graduated or graduated students. The course comes in the sequence of a previous introduction to computational systems security topics, allowing students to gain a deeper comprehension of theoretical and practical dimensions of computer systems and networks security and dependability solutions.

The course aims to expose students with a survey of both the principles and practice on computer systems and network security, starting from theoretical foundations, mechanisms, techniques, security standards and best practices in the design and implementation of security services and protocols for large-scale dependable distributed systems and internetworked applications.

Security design principles and techniques to implement TCP/IP internetworked applications and large scale internet-based services, are particularly emphasized, and complementary trained in the laboratory with programming exercises, practical work-assignments and a final mini-project.

3.3.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas são usadas para apresentação e discussão dos tópicos do programa, com ênfase nos conceitos, fundamentos, paradigmas, técnicas e casos de estudo associados a mecanismos, protocolos e serviços de segurança.

As aulas práticas são utilizadas para enquadramento de exercícios e trabalhos em laboratório (incluindo exercícios de programação, demonstração de ferramentas e sua avaliação experimental), complementados por trabalhos e projetos de avaliação com forte componente de implementação e avaliação experimental.

Componentes de avaliação e seu peso na avaliação:

- 1. Dois testes de frequência (50 %)*
- 2. Trabalhos de avaliação com relatório escrito e apresentação e discussão em laboratório (25%)*
- 3. Projeto de avaliação em grupo, com componente de implementação e análise experimental, podendo abarcar o estudo prático de um tópico de introdução à investigação (25%)*

3.3.7. Teaching methodologies (including evaluation):

Lectures for motivating, presenting and discussing principles, concepts, foundations, paradigms, techniques, case-studies, with the discussion of good practices and explanation of algorithms, protocols and services for computer networks and distributed systems.

Practical classes and lab-exercises (including programming exercises, demonstrations and experimental assessment) are used for training programming techniques and to improve "hand-on" practical skills. Work-assignments and mini-projects are suggested as mandatory practical assessment and optional autonomous work, for the consolidation of theoretical topics.

Assessment components:

- 1. Two intermediate tests (50 % of the final grade)*
- 2. Lab work assignments with required written reports and experimental analysis (25% of the final grade)*
- 3. Group research and implementation project (25% of final grade)*

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Sendo os sistemas computacionais sistemas distribuídos com requisitos da distribuição envolvendo cenários de grande escala, complexidade e critérios de manutenção e gestão descentralizadas, os problemas de segurança são condicionantes fundamentais para a conceção e operação de sistemas informáticos confiáveis. Os critérios de segurança devem ser estabelecidos por definição e especificação, em tempo de conceção dos sistemas e prevendo a sua exposição e operação, ao nível das múltiplas dimensões de suporte de um sistema distribuído. Tal requer um conhecimento profundo e um domínio de detalhe dos princípios e fundamentos teóricos dos mecanismos e serviços de segurança associados aos sistemas e redes de computadores, bem como prática da sua concretização e avaliação experimental.

A unidade curricular conjuga os fundamentos teóricos e a aprendizagem prática de mecanismos, componentes, técnicas, protocolos e aspetos de normalização, tendo em vista uma compreensão de detalhe, conceção, implementação e avaliação de serviços de segurança para redes de computadores, sistemas distribuídos e suas aplicações. O programa conjuga as capacidades de compreensão e sistematização dos conhecimentos teóricos fundamentais com a aquisição de competências práticas orientadas para a conceção,

desenvolvimento e avaliação experimental de protocolos e serviços de segurança, com uma forte vertente de implementação e análise crítica. É particularmente enfatizada a aquisição de conhecimento teórico-prático associado aos princípios, técnicas e normalização de protocolos e serviços de segurança suportados em redes TCP/IP.

Cada um dos objetivos pedagógicos é tratado com base em referências bibliográficas e metas de aprendizagem amplamente utilizados em cursos congêneres de universidades de referência no plano nacional e internacional. A metodologia de ensino exige a participação ativa e individual dos alunos no acompanhamento das aulas teóricas e práticas, sendo esse acompanhamento complementado pelo estudo autónomo de referências apresentadas e discutidas nas aulas teóricas como leituras sugeridas e exercícios práticos propostos.

Aos alunos é exigido um envolvimento individual importante na realização dos exercícios e trabalhos práticos em laboratório, com particular ênfase nos trabalhos de avaliação que requerem uma forte componente de implementação e análise crítica com avaliação experimental.

As leituras sugeridas para aprofundamento e estudo autónomo dos alunos podem abarcar aspetos ou temas de introdução a tópicos de investigação, com enquadramento científico. Estes tópicos serão preferencialmente enquadrados no âmbito do projeto de avaliação final, desenvolvido em grupo. Estes temas podem introduzir problemas de investigação para tratamento e aprofundamento posterior no âmbito de preparação e elaboração de dissertações de mestrado na área da segurança para sistemas distribuídos e redes de computadores.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In this age of universal large scale internetworked computing environments, where interoperability criteria, computing vulnerabilities, failures, hackers, viruses, electronic eavesdropping or electronic frauds can coexist, there is indeed no time at which security does not matter. Security must be established today "by design" and "at design time", requiring a strong and deep knowledge of security and dependability design principles, security services and secure operation practices, as base skills of computer scientists and engineers. The course provides a survey of both the principles and practice on computer systems and network security, starting from theoretical foundations, mechanisms, techniques, security standards and best practices to design and implement security services and protocols for distributed systems and internetworked applications. Security design principles and techniques to implement TCP/IP internetworked applications and large scale internet-based services, are particularly emphasized.

Each of the learning outcomes is carefully treated, using well-known and remarkable bibliographic sources, suggested objectives (as suggested in the reference bibliography) and course syllabus from reference schools and similar courses. The understanding and the comprehensive framework of the main topics is dealt by theoretical classes and lectures, as well as, mandatory assessment of suggested and selected readings for frequency tests and exams.

Students must attain a reasonable level of expertise supported in individual study of those readings, complemented by suggested research papers, optionally used for the topics covered in work-assignments or final mini-projects. Laboratory sessions and programming exercises allow students to be confronted with the practical challenge of the implementation of security protocols and services, as a corollary of the theoretical study on security foundations and mechanisms. Real world implementations, first modelling the adversary conditions and security properties as inputs to implementation requirements and experimental assessment, with the related analysis of implementation results, follow a teaching methodology oriented to a scientific approach of designing effective and proven security solutions.

In the final part of the course, advanced topics in new and emergent proposals are also researched (and tested) as optional group projects, motivating students particularly interested in following a research-work preparation on a MSc thesis security topic.

3.3.9. Bibliografia principal:

Bibliography

• William Stallings, Cryptography and Network Security - Principles and Practice, Prentice Hall, 5th Edition ISBN 13-978-0-13-609704-4, 2011

Other references

• Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security - Private Communication in a Public World, Prentice Hall, ISBN 0-130-46019-2, 2002

• Bruce Schneier, Applied Cryptography - Protocols, Algorithms and Source Code in C, 2nd edition, J. Wiley & Sons, ISBN 0471-12845-7 (v. cloth: acid-free paper) or ISBN 0471-11709-9 (paper: acid-free paper)

• Ross J. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, J. Wiley & Sons, ISBN 0471-38922-6, 2001

• K. P. Birman, Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services, Springer, 2012

Optional research papers or technical reports may be proposed as suggested readings for work-assignments, study surveys or practical labs.

Mapa IV - Síntese de Imagem / Image Synthesis

3.3.1. Unidade curricular:

Síntese de Imagem / Image Synthesis

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Manuel João Toscano Próspero dos Santos (T-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Fernando Pedro Reino da Silva Birra (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Modelação básica 3D e constituição de um «pipeline» de visualização*
- *Modelos de cor e sua razão de ser*
- *Técnicas e modelos de iluminação, locais e globais*
- *Técnicas para animação por computador*
- *Interação com gráficos 3D numa aplicação*

Saber Fazer:

- *Programar grande parte das diversas técnicas lecionadas*
- *Construir aplicações gráficas 3D interativas a nível API*
- *Reutilizar componentes de programação no domínio*
- *Escolher e aplicar a técnica mais adequada à resolução de um problema*

Competências Complementares:

- *Capacidade para analisar sistemas gráficos comerciais*
- *Entendimento dos objetivos das linhas de investigação sobre «rendering»*
- *Maior facilidade de visualização e orientação no espaço, bem como no tempo*
- *Melhor compreensão de áreas relacionadas, como a fotografia, o cinema e a física*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *The 3D viewing pipeline and basic 3D modelling techniques*
- *Colour models*
- *Local and global illumination models and techniques*
- *Computer animation techniques*
- *User interaction with 3D graphics*

Application:

- *Implementing the key techniques taught*
- *Programming 3D interactive applications at the API level of industrial software*
- *Reusability of programming components in the interactive graphics domain*
- *To choose and implement the most appropriate method to solve a problem*

Soft-skills:

- *Ability to analyze graphical systems available on the market*
- *Understanding some goals of research streams about rendering*
- *Increased ease of viewing and orientation in both space and time*
- *Better understanding of related areas such as photography, film and physics*

3.3.5. Conteúdos programáticos:

1. *Modelação B-rep*
2. *Pipeline gráfico 3D*
3. *O paradigma da máquina fotográfica*
4. *Cor acromática e técnicas de meios-tons*
5. *Colorimetria*
6. *Gamas de cor e sistemas de gestão de cores*
7. *Leis da fotometria*
8. *Modelos de iluminação e sombreamento*
9. *Ray-tracing e ray-casting*

10. Radiosidade
11. Efeitos especiais para aumentar o realismo
12. Fundamentos da animação por quadros-chave
13. Programação de aplicações gráficas 3D interativas (OpenGL, X3D and Java3D)

3.3.5. Syllabus:

1. B-rep modeling
2. 3D viewing pipeline
3. The synthetic camera paradigm
4. Achromatic color and halftone techniques
5. Colorimetry
6. Color gamuts and color management systems
7. Photometry laws
8. Illumination models and shading
9. Ray-tracing and ray-casting
10. Radiosity
11. Special effects to improve realism
12. Fundamentals of key-frame animation
13. Programming of 3D interactive graphical applications (OpenGL, X3D and Java3D)

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

O programa inclui os conceitos fundamentais (1, 2, 4, 5, 6 e 7), técnicas e algoritmos para a produção de imagens de síntese com aspeto fotorrealista (8, 9, 10 e 11). Para além disso, é introduzido o paradigma da máquina fotográfica (3) juntamente com algoritmos e técnicas para a interação com o utilizador e a animação gráfica (12 e 13).

A programação de aplicações gráficas 3D interativas utilizando diversas plataformas ao nível API (13) dá ao aluno a oportunidade de conhecer diferentes maneiras de implementar os principais conceitos teóricos e melhorar as suas capacidades de análise no que diz respeito aos sistemas gráficos e ferramentas relacionadas.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus includes the fundamental concepts (1, 2, 4, 5, 6 and 7), techniques and algorithms for the production of synthetic images with a photo-realistic aspect (8, 9, 10 and 11). Moreover, the synthetic camera paradigm is introduced (3) along with algorithms and techniques for user interaction and computer animation (12 and 13).

The programming of 3D interactive graphical applications by using diverse platforms at the API level (13) gives the student the opportunity to know different ways to implement the main theoretical concepts and to improve his/her analysis capabilities regarding graphical systems and related tools.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas, as matérias são explicadas e ilustradas com exemplos de programação. Em geral, todos os resultados principais são mostrados com base numa variedade de meios, incluindo applets.

Nas aulas práticas, os alunos fazem pelo menos dois projetos que abrangem a programação de questões importantes na geração de imagens de síntese e onde eles projetam e implementam a interface gráfica para interagir com as aplicações gráficas 3D correspondentes. Os alunos devem frequentar a maioria das aulas práticas.

A cada aluno será atribuída uma nota individual resultante da avaliação dos projetos de programação e de dois testes intermédios. Este procedimento irá substituir o exame final em caso de aprovação, mas o aluno deve obter uma nota mínima com os dois testes.

3.3.7. Teaching methodologies (including evaluation):

In theoretical classes, matters are explained and illustrated with programming examples. In general, all the main results are shown using a variety of means, including applets.

In practical classes, students do at least two projects covering the programming of important issues in the generation of synthetic images and the design of a suitable GUI to interact with the corresponding 3D graphical applications. Students must attend the majority of practical classes.

Each student will be assigned an individual grade resulting from the evaluation of the programming projects and two mid-term tests. This procedure will substitute the final exam in case of approval, but the student must achieve a minimum grade with the two mid-term tests.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Através das aulas teóricas o aluno é iniciado nos conceitos, técnicas, ferramentas matemáticas e algoritmos que permitem a compreensão da modelação de dados 3D e visualização fotorrealista em computador. A avaliação relacionada é baseada em dois testes intermédios.

As aulas práticas são utilizadas principalmente para a conceção e execução dos projetos, podendo os alunos fazer perguntas de esclarecimento. Esses projetos são aplicações gráficas 3D interativas que usam mais do que uma API atual e refletem o paradigma da máquina fotográfica. É incentivada a reutilização de componentes de programação. A animação por computador, a planificação de movimentos e a simulação física também são objetivos importantes dos projetos.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Throughout lectures the student is introduced to the concepts, techniques, mathematical tools and algorithms allowing the understanding of 3D data modeling and photo-realist rendering in a computer. The related evaluation is based on two mid-term tests.

Practical classes are mainly used for the design and implementation of the projects on which students can ask questions for clarification. These projects are interactive 3D graphical applications using more than one contemporary API and reflecting the synthetic camera paradigm. The reuse of programming components is encouraged. Computer animation, motion planning and physics simulation are also important goals for the projects.

3.3.9. Bibliografia principal:

- J.F. Hughes, A. van Dam, M. McGuire, D. Sklar, J.D. Foley, S.K. Feiner, K. Akeley, "Computer Graphics: Principles and Practice", 3rd Edition, Addison-Wesley Professional (2013), ISBN-13: 978-0321399526
- T. Akenine-Moller, E. Haines, N. Hoffman, "Real-Time Rendering", Third Edition, AK Peters (2008), ISBN-13: 978-1568814247
- D. Hearn, M.P. Baker, W. Carithers, "Computer Graphics with OpenGL, 4th edition, Prentice Hall (2010), ISBN-13: 978-0136053583
- A. Davison, "Pro Java 6 3D Game Development", Apress (2007), ISBN: 1-59059-817-2
- I. Palmer, "Essential Java3d Fast : Developing 3d Applications in Java", Essential Series, Springer-Verlag (2001), ISBN: 1852333944
- R.J. Rost, B.M. Licea-Kane, D. Ginsburg, J. Kessenich, B. Lichtenbelt, H. Malan, M. Weiblen, "OpenGL Shading Language", 3rd Edition, Addison-Wesley Professional (2009), ISBN-13: 978-0321637635
- D. Brutzman, L. Daly, "X3D: 3D Graphics for Web Authors", Elsevier (2007), ISBN: 012088500

Mapa IV - Sistemas de Computação em Cloud / Cloud Computing Systems

3.3.1. Unidade curricular:

Sistemas de Computação em Cloud / Cloud Computing Systems

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Alberto Cardoso e Cunha (T-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Vítor Manuel Alves Duarte (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *os fundamentos dos sistemas de computação em Grid e em Cloud, incluindo os modelos de computação e os paradigmas para a programação de aplicações, para a virtualização de recursos software e hardware e para a organização de serviços sobre as arquiteturas de suporte.*
- *os modelos de programação paralela como o MapReduce para o desenvolvimento de aplicações paralelas que processam grandes volumes de dados*

Saber Fazer:

- *saber identificar as classes de aplicações que beneficiam de sistemas de computação em Grid e Cloud*
- *familiaridade com as abordagens de desenvolvimento de aplicações para computação em Grid e Cloud*
- *capacidade de realizar e avaliar aplicações e serviços sobre plataformas de máquinas virtual distribuídas*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *the principles of Grid and Cloud computing systems, their models and paradigms concerning application development, hardware/software resource virtualization, and service-oriented architectures.*
- *the programming models and environments, such as MapReduce for developing and deploying large-scale data-intensive applications*

Application:

- *ability to identify the main application classes that take advantage from Grid and Cloud computing*
- *learning of the approaches for application development in Grid and Cloud systems*
- *ability to design, implement and experimentally evaluate applications and services supported by distributed virtual machine platforms*

3.3.5. Conteúdos programáticos:

1- Características dos sistemas de Grid e Cloud: grande escala, distribuídos, heterogéneos e dinâmicos. Desempenho, escala e disponibilidade. Virtualização. Abstrações de serviços e recursos.

2- Aplicações: dominadas por computação, dados, interação; paralelismo com grandes volumes de dados; análise dados em grande escala.

3- Computação em Grid: modelos e arquiteturas, gestão de trabalhos, gestão e escalonamento global e local de recursos, programação e execução de aplicações, arquitetura de Grid orientada para serviços.

4- Computação em Cloud: funcionalidades oferecidas como serviços; virtualização; computação sob pedido e conceito de utilidade; Visões do utilizador e do provedor; Plataformas de computação em Cloud: infraestrutura física, camadas de virtualização, gestão de recursos, escalonamento e regulação da carga dinâmica, contratos de garantia de qualidade de serviço; modelos de custo da utilização; Programação baseada em modelos de serviços Cloud e em MapReduce.

3.3.5. Syllabus:

1. Characteristics of Grid and Cloud computing: Large-scale distributed, heterogeneous, dynamic systems. Performance, scalability, availability. Virtualization. User, service, and resource abstractions.

2. Applications: Computing, data, and interaction-centric, massively parallel. Large-scale data analytics.

3. Grid computing: Models and architectures. Job management. Scheduling, global and local resource management. Programming and deploying Grid applications. Towards open Grid service architectures.

4. Cloud computing: Functionalities as a service, virtualization, on-demand, utility computing. User's and Provider's views. Cloud platforms: Physical infrastructure; virtual machine management; dynamic and scalable resource management, service level agreement: load balancing, scheduling and provisioning with quality of service; billing. Cloud programming: Using Cloud service models and MapReduce for application development.

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Estudam-se os fundamentos, os modelos e as arquiteturas dos sistemas de computação paralela e distribuída sobre plataformas de Grid e Cloud e identificam-se as novas classes de aplicações de Grids/Clouds e o seu impacto em diversas áreas disciplinares. As gerações de sistemas de computação são apresentadas numa perspetiva evolutiva, abrangendo Clusters, Grids e Clouds, sendo identificadas as suas abstrações visíveis ao utilizador, bem como as abstrações para gestão de recursos.

Apresentam-se os modelos genéricos característicos destes sistemas e analisam-se as soluções do estado da arte, suportadas por plataformas de Grid e de Cloud. O aluno adquire uma prática laboratorial significativa, baseada na familiarização com os ambientes de suporte baseados em máquinas virtuais distribuídas e os modelos de programação como o MapReduce para o desenvolvimento de aplicações com grandes volumes de dados.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The principles, models and architectures of Grid and Cloud computing systems are studied, jointly with the new application classes and their impact upon multiple application domains. The evolution of computing systems ranging from Clusters and Grids to Clouds is discussed, and their user and system abstractions are identified. The generic computing models of Grids and Clouds, and their state-of-the-art solutions are presented. The student acquires a significant lab practice based on support environments for distributed virtual machines, and programming models such as MapReduce for the development of large data-intensive applications.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é desenvolvido através de sessões teóricas e sessões laboratoriais. Nas primeiras, são apresentados e enquadrados os temas tratados, incentivando-se a perspetiva crítica do aluno, na identificação

dos modelos de computação genéricos dos sistemas tratados, nos seus aspetos comuns e diferenciadores e na sua confrontação com o estudo de casos de sistemas existentes concretos. Essas dimensões são aprofundadas através da realização de um conjunto de trabalhos práticos, de natureza laboratorial, nos quais os alunos desenvolvem e avaliam experimentalmente soluções a nível da conceção de aplicações e da gestão dos ambientes computacionais característicos.

A avaliação assenta em testes presenciais individuais e em trabalhos laboratoriais ao longo do semestre, em grupos, havendo um relatório escrito por cada trabalho e sua discussão oral.

3.3.7. Teaching methodologies (including evaluation):

The course is based on lectures and on a laboratorial component. During the lectures, the themes listed in the syllabus are presented and discussed, in order to stimulate a critical perspective by the student concerning the identification of the studied computing models, in their common and their distinguishing aspects. The discussion is also illustrated through the analysis of the case studies of existing systems and solutions. Furthermore the above dimensions are elaborated through the experimental practical labs, requiring the students to develop, deploy and evaluate solutions for selected application problems, and ensuring an efficient management of the computing environments.

The student evaluation is based on individual written tests, and on lab assignments performed by groups of students along the course, with a written report and oral discussion per assignment.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

A aquisição do conhecimento das características e dos modelos de computação em Grid/Cloud exige uma análise crítica que permita compreender estes novos modelos, como evolução face aos conceitos dos sistemas paralelos e distribuídos tradicionais. Esta reflexão é conduzida ao longo das sessões teóricas, baseadas na exposição e discussão dos conceitos genéricos e na análise de casos concretos, e em que a interação entre o docente e os alunos é fundamental pois esta discussão abre o caminho para a identificação dos novos paradigmas, a nível das aplicações e das estratégias internas aos sistemas em causas, as quais irão ser desenvolvidas pelo aluno para realizar os trabalhos laboratoriais. O aluno é, assim, estimulado para ser pró-ativo na procura de soluções que lhe permitam resolver os problemas de conceção e implementação que lhe são colocados.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In order to reach the required knowledge on the characteristics of the Grid and Cloud computing models and systems, a critical perspective is needed so that the student may understand how those new models represent an evolution of the more traditional parallel and distributed systems. Such analysis is conducted during the lectures, through the presentation of the course topics, and their discussion with the students. This actively involves the student in the identification of the novel paradigms, and their distinguished characteristics, both concerning their practical applications and the internal strategies underlying the implementation of those systems. The student is motivated to reach such an understanding and to be proactive, in order to achieve the more effective design and implementation solutions to solve the practical laboratory assignments.

3.3.9. Bibliografia principal:

- K. Hwang, G. C. Fox, J. J. Dongarra, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*, Morgan-Kaufmann, 2012.
- *Cloud Computing: Principles, Systems, and Applications*, N. Antonopoulos, L. Gillam (Eds), Springer, 2010.
- *The Grid2: Blueprint for a New Computing Infrastructure*, I. Foster, C. Kesselman (Eds), Morgan-Kaufmann, 2003.
- K. P. Birman, *Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services*, Springer, 2012.

Mapa IV - Sistemas de Computação Móvel e Ubíqua / Mobile and Ubiquitous Computing

3.3.1. Unidade curricular:

Sistemas de Computação Móvel e Ubíqua / Mobile and Ubiquitous Computing

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Augusto Legatheaux Martins (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Nuno Manuel Ribeiro Preguiça (Regente) (T-28h)

Sérgio Marco Duarte (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Esta é uma unidade curricular (UC) opcional sobre sistemas de computação móvel e ubíqua. Esta UC introduz as características fundamentais dos ambientes de computação móvel e ubíqua, assim como os modelos e técnicas usadas no desenvolvimento de aplicações para estes ambientes. Como pré-requisito, os estudantes devem possuir conhecimentos de programação e sistemas distribuídos.

Saber:

- *Características dos ambientes de computação móvel e ubíqua*
- *Arquiteturas, modelos e técnicas de programação para estes ambientes*
- *Conhecimentos fundamentais de redes móveis*
- *Técnicas de localização*
- *Problemas e soluções específicas para aplicações móveis (interfaces, gestão de dados e adaptação ao contexto)*
- *Problemas e técnicas de sistemas de sensoriamento*

Saber Fazer:

- *Desenho e implementação de sistemas móveis*
- *Implementação de soluções para lidar com aspetos específicos relacionados com a mobilidade*
- *Desenho e implementação de sistemas ubíquos*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

This is an optional course on mobile and pervasive computing systems. It provides the basic knowledge on the characteristics of mobile and pervasive computing, models and techniques used in the development of applications targeting these environments. As prerequisites students should have previous acquaintance with programming and distributed systems.

Knowledge:

- *Characteristics of mobile and pervasive computing*
- *Architectures, models and techniques for programming mobile and pervasive computing system*
- *Basic knowledge on mobile networks*
- *Location techniques*
- *Specific problems and solutions for mobile applications (interfaces, data management, context awareness)*
- *Problems and techniques for sensing systems*

Application:

- *Design and implement mobile systems*
- *Implement solutions to address mobility specific issues*
- *Design and implement pervasive systems*

3.3.5. Conteúdos programáticos:

- 1. Introdução, exemplos e desafios*
- 2. Programação de sistemas móveis*
 - 2.1 Modelos e arquiteturas*
 - 2.2 Frameworks nativos*
 - 2.3 Soluções baseadas na web*
- 3. Redes sem fios*
 - 3.1 Redes sem fios*
 - 3.2 Mobilidade*
- 4. Localização*
 - 4.1 Conceitos de localização*
 - 4.2 Técnicas de localização*
- 5. Aspetos específicos da computação móvel*
 - 5.1 Interfaces*
 - 5.2 Gestão de dados*
 - 5.3 Adaptação ao contexto*
- 6. Introdução à computação ubíqua*
 - 6.1 Dispositivos*
 - 6.2 Modelos a arquiteturas*
- 7. Sistemas de sensoriamento*
 - 7.1 Redes de sensores*

7.2 Sensoriamento participativo

3.3.5. Syllabus:

1. *Introduction, examples, and challenges.*
2. *Programming mobile systems*
 - 2.1 *Models and architectures*
 - 2.2 *Native frameworks*
 - 2.3 *Web-based solutions*
3. *Wireless networks*
 - 3.1 *Wireless networks*
 - 3.2 *Mobility*
4. *Location*
 - 4.1 *Location concepts*
 - 4.2 *Location techniques*
5. *Mobile Computing Specific Issues*
 - 5.1 *Interfaces*
 - 5.2 *Data management*
 - 5.3 *Context-awareness*
6. *Introduction to pervasive computing*
 - 6.1 *Devices*
 - 6.2 *Models for pervasive computing*
7. *Pervasive Sensing*
 - 7.1 *Sensor networks*
 - 7.2 *Participatory sensing*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A UC visa fornecer um conjunto de conhecimentos e competências que lhe permitam desenhar e desenvolver aplicações em sistemas de computação móvel e ubíqua.

Para atingir esse objetivo, a UC está estruturada em duas partes: a primeira aborda ambientes de computação móvel tradicional e a segunda estende o âmbito à computação ubíqua.

No contexto da computação móvel, apresenta as aproximações gerais de programação de aplicações nestes ambientes, seguida de aspetos específicos dos ambientes móveis, incluindo redes sem fios, sistemas de localização, interfaces, gestão de dados e adaptação ao contexto.

No contexto da computação ubíqua, estudam-se os aspetos relacionados com a programação e os sistemas de sensoriamento.

Na componente prática, os conceitos estudados na parte teórica serão colocados em prática em aplicações móveis, permitindo aos estudantes ganhar experiência no desenvolvimento deste tipo de sistemas.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

This course aims to provide students with a set of knowledge and know-how that allows them to develop applications for mobile and pervasive computing environments.

To reach this goal, the course includes two parts focusing on mobile computing and pervasive computing respectively.

In the context of mobile computing, it starts by addressing the general solutions for programming mobile systems, and continues by addressing mobile systems issues, including wireless networks, location techniques, interfaces, data management and context adaptation.

In the context of pervasive systems, the course will focus on programming approaches and sensing systems.

In the labs, the topics studied in the lectures will be exercised in the development of mobile systems focusing on different aspects, allowing students to gain experience in programming this type of systems.

3.3.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas expõe-se e discute-se o programa da UC, apresentando os vários tópicos devidamente enquadrados com exemplos de sistemas existente que implementam as soluções estudadas.

As aulas práticas acompanham os dois projetos da UC, fornecendo aos alunos experiência no desenvolvimento de sistemas móveis usando as técnicas estudadas. Os projetos abordam tópicos específico da matéria abordada nas aulas teóricas. Nas aulas práticas apresentam-se exemplos e inicia-se a resolução dos projetos, e presta-se apoio à sua realização.

A avaliação será baseada nos seguintes componentes: dois testes (30% cada); dois projetos de programação, resolvidos em grupo (20% cada).

3.3.7. Teaching methodologies (including evaluation):

In lectures, the key points of the syllabus are presented, using examples of existing systems whenever possible. This presentation should be complemented with studying the recommended materials.

The laboratory classes target the presentation and resolution of two programming projects, providing students with experience in using the studied techniques for programming mobile systems. Projects will focus on different aspect of the studied topics. Examples will be presented in laboratory classes, where the projects will be partially executed.

The course grade will be computed based on the following components: two quizzes (30% each) and two programming projects solved in team (20% each).

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As metodologias de ensino têm como objetivo permitir conhecer os modelos e técnicas usadas na concepção e criação de sistemas móveis, assim como as limitações tecnológicas que se colocam nos ambientes de computação móvel e ubíqua. Tal é conseguido através da exposição dos conceitos e técnicas fundamentais nas aulas teóricas e sua consolidação e colocação em prática nas aulas práticas, através da apresentação de exemplos e realização de dois projetos de programação.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The teaching methodologies have the goal of allowing students to know the models and techniques used in the developments of mobile systems, as well as the technological limitations that exist in such environments. This is achieved by presenting the fundamental concepts and techniques in lectures. In labs, the knowledge obtained in lectures is consolidated and exercised by the presentation of examples and the execution of two programming projects.

3.3.9. Bibliografia principal:

Main Reference:

Frank Adelstein, Sandeep Gupta, Golden Richard, Loren Schwiebert, Fundamentals of Mobile and Pervasive Computing, McGraw-Hill, 2005 ISBN 0071412379

The main reference is complemented with a selection of articles on specific topics.

Mapa IV - Sistemas Multi-Agente / Multi Agent Systems

3.3.1. Unidade curricular:

Sistemas Multi-Agente / Multi Agent Systems

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

José Júlio Alves Alferes (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

João Alexandre Carvalho Pinheiro Leite (Regente) (T-28h)

Jorge Carlos Ferreira Rodrigues da Cruz (P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- Arquiteturas de Agentes*
- A linguagem de Programação para Agentes Jason*
- Mecanismos e estratégias de decisão individual em sistemas multiagente.*
- Mecanismos e estratégias de negociação, decisão distribuída e colaboração.*

Saber Fazer:

- Analisar e modelar um problema usando um sistema Multi-Agente*
- Implementar um Sistema Multi-Agente em Jason.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Agent Architectures*
- *The Agent Oriented Programming Language Jason*
- *Mechanisms and Strategies for individual decision making in Multi-Agent Systems*
- *Mechanisms and Strategies for negotiation, distributed decision making and collaboration in Multi-Agent Systems.*

Application:

- *Analyse and model a problema using a multiagent system*
- *Implement a multiagent system in Jason*

3.3.5. Conteúdos programáticos:

1. *Introdução aos sistemas multiagente*
2. *Arquiteturas de Agentes*
 - 2.1. *Agentes Inteligentes*
 - 2.2. *Agentes de raciocínio dedutivo*
 - 2.3. *Raciocínio prático*
 - 2.4. *Arquiteturas reativas e híbridas*
3. *Linguagens de programação orientadas a agentes*
4. *Interação entre agentes e comunicação*
 - 4.1. *Interações multiagente*
 - 4.2. *comunicação entre agentes*
 - 4.3. *encontros multiagente*
5. *Alcançar acordos*
 - 5.1. *Decisão racional distribuída*
 - 5.2. *Negociação*
 - 5.3. *Argumentação*
6. *Cooperação*
7. *Engenharia de Software orientada a Agentes*

3.3.5. Syllabus:

1. *Introduction to Multi-Agent Systems*
2. *Agent Architectures*
 - 2.1. *Intelligent Agents*
 - 2.2. *Deductive Reasoning Agents*
 - 2.3. *Practical Reasoning*
 - 2.4. *Reactive and Hybrid Architectures*
3. *Agent-oriented Programming Languages*
4. *Agent Interaction and Communication*
 - 4.1. *Multi-Agent Interactions*
 - 4.2. *Agent Communication*
 - 4.3. *Multi-Agent Encounters*
5. *Reaching Agreements*
 - 5.1. *Distributed Rational Decision Making*
 - 5.2. *Negotiation*
 - 5.3. *Argumentation*
6. *Cooperation*
7. *Agent Oriented Software Engineering*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os conteúdos programáticos seguem de perto os objetivos de conhecimento estabelecidos, sendo assim clara a coerência entre ambos.

As aptidões objetivo da unidade curricular decorrem da execução do programa em atividades práticas laboratoriais.

Os conteúdos programáticos relativos à Engenharia de Software orientada a Agentes, conjuntamente com os relativos às várias arquiteturas de agentes, interação, alcance de acordos e cooperação são coerentes com o objetivo de saber analisar e modelar um problema como um sistema multiagente. Os conteúdos programáticos relativos à linguagem Jason são coerentes com o objetivo correspondente.

Todas as competências são também fomentadas pela forma de avaliação da unidade curricular, quer pelos testes/exame quer pelo trabalho prático.

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus follows very closely the knowledge learning outcomes established for the course, so it is clear the

coherence between the two.

The know-how learning outcomes stem from the exercise of the program in lab sessions.

The syllabus items corresponding to Agent Oriented Software Engineering, together with those related to Agent Architectures, Agent Interaction, reaching agreements and cooperation are all coherent with the know-how objective of analysing and modelling a problem as a multiagent system. The syllabus item corresponding to the Agent Programming Language Jason is coherent with the corresponding implementation know-how objective. All learning outcomes are promoted through the evaluation mechanisms, both by the tests/exam and by the projects.

3.3.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas e práticas. Nas primeiras são lecionados os conceitos e técnicas relevantes. Nas aulas práticas são resolvidos problemas, feitas experiências em laboratório de computadores, e (parcialmente) desenvolvido o trabalho prático usando Jason.

A avaliação inclui:

•2 testes individuais teóricos (ou um exame de recurso), onde é avaliado o conhecimento que os alunos adquiriram dos conceitos e características das técnicas lecionadas.

•1 trabalho prático, com duas componentes classificadas individualmente, elaborado por 2 estudantes, que consiste no desenho, análise, e implementação de uma equipa de agentes para participar no concurso de programação de agentes (<http://multiagentcontest.org/>). A entrega e avaliação do trabalho prático é feita em duas fases: numa primeira fase é entregue o desenho do sistema multiagente; numa segunda fase é entregue o projeto completo. A avaliação do projeto final inclui uma competição entre os vários sistemas apresentados.

3.3.7. Teaching methodologies (including evaluation):

The syllabus is taught in theoretical and laboratory classes. In the former, the main concepts and techniques are addressed.

The laboratory classes are dedicated to solving problems, and experimenting the various concepts of the syllabus using the Jason, and (partially) developing the project.

The evaluation includes:

•2 midterm individual tests (or an exam) where the knowledge acquired by the students on the concepts and techniques is assessed.

•1 practical project, with two separately assessed components, developed in groups of 2 students, consisting of the design, analysis and implementation of a team of agents to participate in the agent programming competition (<http://multiagentcontest.org/>).

The project is developed/evaluated in two steps: first the design of the multiagent system; second, the complete project. The evaluation of the final project includes a competition between all submitted systems.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular identificam a) a aquisição de conhecimento sobre técnicas e mecanismos relevantes no contexto de sistemas multiagente; b) a aquisição de conhecimento sobre desenho e implementação de sistemas multiagente; c) a aplicação prática dos conhecimentos adquiridos

O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição das várias arquiteturas de agentes, as várias técnicas que os agentes podem usar, quer para tomar decisões na presença de adversários (decisões individuais), quer para interagir com outros agentes, e.g. através de negociação, argumentação e eventual cooperação, que são os ingredientes necessários para o desenho de um sistema-multi-agente.

O objetivo b) é atingido com a exposição sobre os métodos de engenharia de software orientada a agentes.

O objetivo c) é fundamentalmente atingido através da aprendizagem da linguagem de programação Jason e da sua utilização nas aulas práticas na resolução de exercícios.

O trabalho prático tem um papel fundamental na aplicação integrada de todas estas competências, no desenho e implementação de um sistema multiagentes. O facto de o trabalho consistir no desenvolvimento de uma equipa para participar num concurso contribui para estimular os alunos e assim, para o atingir dos objetivos. As matérias relativas ao objetivo a) são avaliadas em testes individuais. As matérias relativas aos objetivos b) e c) são avaliadas no trabalho prático.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

The objectives of the course identify a) the acquisition of knowledge about techniques and mechanisms relevant within the context of multiagent systems; b) the acquisition of knowledge about the design and implementation of a multiagent system, b) the practical application of acquired knowledge.

Objective a) is mainly achieved in lectures with the presentation of the various agent architectures, the various techniques agents can use both to (individually) decide in the presence of adversaries and to interact with other agents e.g. through negotiation, argumentation and eventual cooperation, which are the necessary ingredients

to the design of a multiagent system.

Objective b) is achieved with the presentation of the methods from agent oriented software engineering.

Objective c) is achieved through the presentation, in classes, of the Agent Oriented Programming Language Jason, and its use in the practical classes to solve exercises.

The practical project has a fundamental role in the joint application of all the learnt techniques in the design and implementation of a multiagent system. The fact that the project consists in the development of a team to participate in the Agent Programming Contest contributes to stimulate students hence contributing to achieving the objectives.

Matters relating to the objectives a) are evaluated through individual tests. Matters relating to objective b) and c) are evaluated through the practical project.

3.3.9. Bibliografia principal:

Livros de Texto / Text Books

- *M Wooldridge. An Introduction to Multiagent Systems. 2nd Edition, John Wiley, 2009.*
- *G Weiss (ed) Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999.*
- *Y. Shoham and K. Leyton-Brown, Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations. Cambridge University Press, 2009.*

Mapa IV - Técnicas Avançadas de Desenho de Algoritmos / Advanced Algorithm Design Techniques

3.3.1. Unidade curricular:

Técnicas Avançadas de Desenho de Algoritmos / Advanced Algorithm Design Techniques

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Luís Manuel Marques da Costa Caires (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

*Margarida Paula Neves Mamede (Regente) (T-28h),
Fernanda Maria Barquinha Tavares Vieira Barbosa (P-28h)*

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Aprofundar o conhecimento sobre duas técnicas de desenho de algoritmos: estratégias greedy e programação dinâmica.*
- *Definir algumas classes de complexidade e compreender alguns problemas em aberto.*
- *Reconhecer os problemas das classes NP e PSPACE.*
- *Conhecer alguns métodos gerais para conceber algoritmos de aproximação e algoritmos aleatórios.*

Saber Fazer:

- *Conceber e analisar um algoritmo aplicando, quer uma estratégia greedy (quando existe), quer programação dinâmica.*
 - *Provar que um problema é NP-completo ou PSPACE-completo.*
 - *Conceber e analisar um algoritmo de aproximação e um algoritmo aleatório para um problema concreto.*
- Selecionar as técnicas apropriadas a um problema.*

Competências Complementares:

- *Capacidade de investigação e autonomia.*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Be familiar with two algorithm design techniques: greedy strategies and dynamic programming.*
- *Define some complexity classes and understand some open problems.*
- *Recognize NP- and PSPACE-problems.*
- *Know some general methods for designing approximation algorithms and randomized algorithms.*

Application:

- *Design and analyse a greedy algorithm (when there is an applicable strategy) and a dynamic programming algorithm.*
- *Make NP- and PSPACE-completeness proofs.*
- *Design and analyse an approximation algorithm and a randomized algorithm for a real-world problem.*
- *Select the suitable techniques to solve a problem.*

Soft-skills:

- *Skills in research and autonomy.*

3.3.5. Conteúdos programáticos:

1. Técnicas avançadas de desenho de algoritmos.

- *Algoritmos greedy.*
- *Programação dinâmica.*

2. Introdução à Teoria da Complexidade.

- *As classes P, FP, NP, PSPACE e EXPTIME.*
- *Os afixos co, difícil e completo.*
- *Redução de problemas.*
- *Alguns problemas em aberto.*

3. Algoritmos de aproximação.

- *Rácio de aproximação.*
- *Estratégias greedy.*
- *Os métodos primal-dual e primal.*
- *Esquemas de aproximação.*

4. Algoritmos aleatórios.

- *Análise probabilística.*
- *Os métodos de Monte Carlo e de Las Vegas.*
- *As classes RP e ZPP.*

3.3.5. Syllabus:

1. Advanced algorithm design techniques.

- *Greedy algorithms. Dynamic programming.*

2. Introduction to the Theory of Complexity.

- *The classes P, FP, NP, PSPACE, and EXPTIME.*
- *The affixes co, hard and complete.*
- *Problem reductions.*
- *Some open problems.*

3. Approximation algorithms.

- *Approximation ratio.*
- *Greedy strategies.*
- *The primal-dual and the primal methods.*
- *Approximation schemes.*

4. Randomized algorithms.

- *Probabilistic analysis.*
- *The Monte Carlo and the Las Vegas methods.*
- *The classes RP and ZPP.*

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular podem ser sintetizados da seguinte forma.

1. Aprofundar o conhecimento sobre algoritmos greedy e programação dinâmica: *Estas técnicas são introduzidas no ponto 1 dos conteúdos programáticos, com vários exemplos, e aplicadas nos pontos 3 e 4, onde são selecionadas para desenhar alguns algoritmos de aproximação ou aleatórios.*

2. Conhecer as principais classes de complexidade, em particular, a dos problemas NP-completos: *Isto é abordado pela primeira vez no ponto 2 e é retomado com frequência nos pontos 3 e 4, porque muitos problemas estudados são NP-difíceis.*

3. Conceber e analisar algoritmos de aproximação e algoritmos aleatórios para problemas concretos: *Os pontos 3 e 4 cobrem esta matéria com muitos exemplos, que ilustram e permitem aplicar os diferentes métodos estudados. O rácio de aproximação e a análise probabilística são necessários para se poderem analisar (avaliar) os algoritmos desenvolvidos.*

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

The unit objectives can be summarized in the following way.

1. Be familiar with greedy algorithms and dynamic programming: *These techniques are introduced in syllabus topic 1, with several examples, and applied in topics 3 and 4, where they are selected to design some approximation or randomized algorithms.*

2. Know the main complexity classes, in particular, that of NP-complete problems: *This is presented for the first time in topic 2 and often brought back in topics 3 and 4, because many studied problems are NP-hard.*

3. Design and analyse approximate algorithms and randomized algorithms for real-world problems: *Topics 3 and*

4 cover this material with many examples, which illustrate and allow all studied methods to be applied. The approximation ratio and the probabilistic analysis are needed to be able to analyse (evaluate) the developed algorithms.

3.3.7. Metodologias de ensino (avaliação incluída):

O ensino consiste na exposição da matéria em aulas teóricas e na resolução de problemas em aulas práticas de laboratório. No laboratório, os alunos identificam as classes a que os problemas pertencem e desenham, analisam e implementam algoritmos.

A avaliação é composta por dois trabalhos e dois testes. O primeiro trabalho consiste na resolução de um problema de um concurso de programação, com uma estratégia greedy ou com programação dinâmica, e o segundo num estudo comparativo, teórico e experimental, de soluções alternativas para um dado problema. Em ambos os casos, os alunos elaboram um relatório e há uma discussão. Se a média dos testes não for positiva, os alunos podem realizar um exame final. Os testes e o exame são com consulta.

Condição para obter aprovação: $NotaP \geq 9.5$ e $NotaT \geq 9.5$, onde $NotaP$ é uma média ponderada das notas dos trabalhos e $NotaT$ é a média das notas dos testes ou a nota do exame.

3.3.7. Teaching methodologies (including evaluation):

There are two lectures and a lab session each week. In the laboratory, students identify the classes to which problems belong and design, analyse and implement algorithms.

Assessment comprises two programming projects and two tests. The first programming project consists in solving a programming contest problem, with a greedy strategy or dynamic programming, and the second in a theoretical and experimental comparative study of alternative solutions for a given problem. In both cases, students write a report and there is a discussion. If the mean of the test grades is less than 9.5 (out of 20), students can do a final exam. The tests and the exam are open-book.

Condition to succeed: $Pgrade \geq 9.5$ and $Tgrade \geq 9.5$, where $Pgrade$ is a weighted average of the project grades and $Tgrade$ is the mean of the test grades or the exam grade.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Nas aulas práticas, os alunos analisam e resolvem problemas concretos, utilizando as técnicas e os métodos dados nas aulas teóricas. Na fase de análise, os alunos começam por classificar o problema, identificando a sua classe de complexidade e o seu tipo. Depois, selecionam a técnica de desenho (ou, nalguns casos, o método) que melhor se adequa àquele caso. Na fase de resolução, os alunos desenharam o algoritmo, aplicando a técnica previamente escolhida, analisam-no e implementam-no. Todas as escolhas têm de ser justificadas.

Nas componentes de avaliação teóricas (testes e exame), os alunos são confrontados com questões e problemas que permitem avaliar a compreensão dos conceitos envolvidos na unidade curricular, a capacidade para efetuar escolhas apropriadas e a aptidão para desenhar e analisar algoritmos.

No primeiro trabalho, os alunos resolvem um problema de um concurso de programação, aplicando uma estratégia greedy ou programação dinâmica. Para além destes problemas serem sempre desafios interessantes, como estão especificados como problemas do mundo real, é mais difícil identificar a forma de os atacar (ou seja, a técnica de desenho de algoritmos que deve ser escolhida). Acresce que estes problemas só se consideram resolvidos quando o programa é aceite por um sistema automático de teste (Mooshak), o que obriga a desenhar e analisar adequadamente o algoritmo antes de o implementar, para que os tempos de execução e a memória requerida não excedam os limites permitidos. Neste caso, os alunos elaboram um relatório sobre o desenho, a análise e a implementação do algoritmo desenvolvido, especificando e justificando todas as opções tomadas.

No segundo trabalho, os alunos escolhem o problema que pretendem resolver e efetuam um estudo comparativo, teórico e experimental, de soluções alternativas para esse problema, utilizando técnicas e métodos lecionados. O relatório apresenta e classifica o problema, identifica os vários modelos teóricos adotados e descreve a sua aplicação, inclui a análise e a implementação dos respetivos algoritmos e contém uma análise dos resultados obtidos experimentalmente. Com este trabalho, pretende-se incentivar a autonomia e desenvolver a capacidade de investigação dos alunos.

Ao realizar ambos os trabalhos, os alunos desenvolvem a aptidão para selecionar e aplicar as técnicas apropriadas, a aptidão para conceber e avaliar soluções e a capacidade de comunicação escrita.

Como em qualquer processo de aprendizagem, é fundamental que os alunos recebam feedback sobre o trabalho que vão realizando. Por esse motivo, os docentes identificam os pontos fortes e fracos de cada resolução, no final das aulas práticas ou logo após os trabalhos terem sido avaliados.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

In lab sessions, students analyse and solve concrete problems, making use of the techniques and methods presented in the lectures. In the analysis phase, students start by classifying the problem, identifying its

complexity class and its type. Then, they select the design technique (or, in some cases, the method) that is more appropriate in that case. In the solution phase, students design the algorithm, applying the chosen technique, and perform its analysis and its implementation. All choices have to be justified.

The questions and problems posed in tests and exams allow the assessment of (1) the understanding of the concepts involved in the unit, (2) the ability to make suitable choices, and (3) the ability to design and analyse algorithms.

In the first programming project, students solve a programming contest problem, applying either a greedy strategy or dynamic programming. Apart from these problems always being interesting challenges, being stated as real-world problems makes it more difficult to identify the way to tackle them (that is, the algorithm design technique that should be chosen). Besides, since a programming contest problem is considered to be solved only when the program is accepted by an automatic judging system (Mooshak), the algorithm has to be suitably designed and analysed before being implemented so that running times and memory requirements do not exceed the allowed limits. In this case, students write a report on the design, analysis and implementation of the developed algorithm, specifying and justifying all choices made.

In the second programming project, students choose the problem they want to solve and perform a theoretical and experimental comparative study of alternative solutions for that problem, applying techniques and methods studied. The report presents and classifies the problem, identifies the theoretical models adopted, describes their application, includes the analysis and the implementation of the corresponding algorithms, and contains an analysis of the experimental results. This aims at improving autonomy and research skills.

By doing both programming projects, students develop the ability to select and apply the appropriate techniques, the ability to create and evaluate solutions, and skills in writing communication.

As in any learning process, students have to receive feedback on their work. For this reason, teachers identify the strong and the weak points of each solution, at the end of lab sessions or just after the evaluation of the programming projects.

3.3.9. Bibliografia principal:

Main References:

1. Jon Kleinberg and Éva Tardos. *Algorithm Design*. Addison-Wesley, 2006.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3rd edition)*. The MIT Press, 2009.

Complementary References:

3. Anany Levitin. *Introduction to The Design and Analysis of Algorithms (3rd edition)*. Addison-Wesley, 2011.
4. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, 1979.

Mapa IV - Tecnologias de Informação Geográfica / Geographic Information Technology

3.3.1. Unidade curricular:

Tecnologias de Informação Geográfica / Geographic Information Technology

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Nuno Correia (apenas Responsável não tem horas de contacto)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

Maria Armada Simenta Rodrigues Grueau (Regente) (T-28h; P-28h)

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

1. *Conhecer a natureza específica da Informação Geográfica*
2. *Como se representa computacionalmente o espaço através de SIG*
3. *Associar modelos de representação espacial a casos de estudo específicos*
4. *Conhecer o ciclo de desenvolvimento de uma aplicação baseada em SIG*

Saber Fazer:

5. *Utilizar as potencialidades de várias tecnologias de informação geográfica*
6. *Desenvolver um projeto SIG de pequena dimensão*

Competências Complementares:

7. *Explorar capacidades de gestão de tempo e organização*
8. *Trabalho de equipa*

9. *Apresentação escrita e oral de trabalhos desenvolvidos*
10. *Saber reconhecer e avaliar as necessidades de utilizadores*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

1. *To Know the specific nature of GI*
2. *To know how to digitally represent space through GIS*
3. *To associate spatial representation models to specific case-studies*
4. *To know the development cycle of a GIS-based application*

Application:

5. *To use the potential of GI Technology*
6. *To develop a small dimension GIS project*

Soft-skills:

7. *Time management and organization*
8. *To present work in oral and written form*
9. *To recognize and evaluate users' needs*

3.3.5. Conteúdos programáticos:

A. Introdução à Informação Geográfica

- *Natureza e funcionalidade da IG*

B. Conceitos espaciais fundamentais

- *Espaço Euclidiano, Conjuntos, Topologia, Redes, Espaços Métricos*

C. Modelos de representação de Informação Geográfica

- *Modelos baseados em propriedades*
- *Modelos baseados em entidades*

D. Estruturas de dados para IG

- *Representações para modelos de propriedades*
- *Representações para modelos de entidades (objetos)*
- *Formatos Vetoriais versus Formatos Raster*
- *Representações para redes*

E. Arquiteturas de SIG

- *Arquiteturas Híbridas, integradas, compostas*
- *Arquiteturas Distribuídas*
- *Sistemas baseados na localização*

F. Interfaces em SIG

- *Interfaces Cartográficas*
- *Geovisualização*
- *Especificidade do desenvolvimento de interfaces para SIG*

G. Recuperação de Informação Geográfica

3.3.5. Syllabus:

A. Introduction to Geographic Information (GI)

- *Nature and functionality of GI*

B. Fundamental spatial concepts

- *Euclidean space, Sets, Topology, Networks, Metric Spaces*

C. GI Representation Models

- *Field-based models*
- *Entity-based models*

D. GI Data structures

- *Representing field-based models*
- *Representing entity-based models (Objects)*
- *Vetor and Raster formats*
- *Network representation*

E. GIS architectures

- *Hybrid, integrated and composed architectures*
- *Distributed architectures*
- *Location-based systems*

F. User-interfaces for GIS

- *Cartographic interfaces*
- *Geo-visualization*
- *Developing interfaces for GIS*

G. Geographic Information Retrieval

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

- *O objetivo 1 é endereçado nos capítulos A e B do programa.*
- *O objetivo 2 é endereçado no capítulo D do programa.*
- *O objetivo 3 é endereçado no capítulo C do programa.*
- *Os capítulos E,F,G do programa têm relevância para o objetivo 4.*
- *Todos os capítulos podem ter relevância para o objetivo 5.*

3.3.6. Demonstration of the syllabus coherence with the curricular unit's intended learning outcomes:

- *Objective 1 is addressed in chapters A and B of the syllabus.*
- *Objective 2 is addressed in chapter D of the syllabus.*
- *Objective 3 is addressed in chapter C of the syllabus.*
- *Chapters E, F and G are of relevance in objective 4.*
- *All chapters are of relevance in objective 5.*

3.3.7. Metodologias de ensino (avaliação incluída):

Os conteúdos teóricos são endereçados em aulas teóricas, sendo instanciados com exercícios e exemplos nas aulas práticas que vão construindo um corpo de técnicas para o desenvolvimento de trabalho prático. Esta componente é avaliada através de 3 mini-testes, valendo cada um 25% da nota (NT).

Os alunos têm que desenvolver um trabalho prático, em grupos, em que concebem um sistema que contempla um conjunto de requisitos fundamentais em sistemas georreferenciados. Este trabalho deve ser objeto de escrita de relatório e apresentação oral. A nota do trabalho vale 25% da nota (NP).

Para obter aprovação à disciplina, os alunos devem conseguir uma nota mínima de 9,5 (em 20) em NT e NP. Em época de recurso, a nota dos três mini-testes é substituída pela nota do exame, que vale 75% (NT) da nota final.

3.3.7. Teaching methodologies (including evaluation):

Theoretical issues are introduced in lectures, and instantiated in exercises and examples in labs, as the students build a body of techniques to support the development of the practical assignment. This component is evaluated through 3 tests, each one worth 25% of the final grade (NT).

Students must develop a group assignment in which they conceive a system contemplating fundamental requirements in geo-referenced systems, and which will provide access to the exam. These students must write a report describing their system and present their work. The assignment grade is worth 25% of the final grade (NP).

In the appeal season, the tests grade is replaced by the exam grade, which is worth 75% (NT) of the final grade. A student is approved if both NP and NT are greater or equal to 9.5 points out of 20.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

As aulas teóricas seguem todo o programa, tendo em conta os objetivos 1 a 4.

Nas aulas práticas são necessários os conhecimentos obtidos nas aulas teóricas (objetivos 1 a 4) mas os exercícios e exemplos considerados estão diretamente relacionados com o objetivo 5, construindo os alunos um conjunto de capacidades de manipulação de ferramentas que lhe permitem depois atingir o objetivo 6. O desenvolvimento do trabalho prático implica o estudo de um domínio de aplicação, assim como a avaliação das necessidades dos utilizadores nesse contexto (objetivo 10).

Durante o desenvolvimento do trabalho prático, os alunos usam os conhecimentos adquiridos nas aulas teóricas e necessitam de ser conhecedores das ferramentas usadas nas aulas práticas. No entanto, precisam também de ser capazes de gerir o tempo e organizarem-se (objetivo 7) . O trabalho prático também implica a escrita de uma relatório e uma apresentação oral (objetivo 8).

Finalmente, nos testes ou exame, são endereçados os objetivos 1 a 4 e 7.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Lectures go over the content of the syllabus, addressing objectives 1 through 4.

Labs take into account what has been introduced in the lectures, thus addressing the objectives 1 through 4.

Moreover, during the labs, exercises and examples address objective 5 and build up to endow students with tools to achieve objective 6.

The development of the practical assignment involves defining an application domain and evaluating users' needs in that context (objective 10).

While developing the practical assignment, the students must be aware of the content provided in the lectures

and be proficient in the use of the tools used in the labs, but they must also be able to manage time and be organized (objective 7). The practical assignment also involves writing a report and orally presenting their work, and this is related with objective 8.

In the tests or exam, objectives 1-4 and 7 are addressed.

3.3.9. Bibliografia principal:

- *Worboys, M. and Duckham, M. GIS: A Computing Perspective, 2nd Edition, CRC Press, 2004.*
- *Longley, P.A., Goodchild, M. F., Maguire, D. J., Rhind, D. W, Geographic Information Systems and Science , 2nd Edition, Wiley, 2005.*
- *Matos, J., Fundamentos de Informação Geográfica, 4ª Edição, LIDEL.*

Mapa IV - Web Semântica / Semantic Web

3.3.1. Unidade curricular:

Web Semântica / Semantic Web

3.3.2. Docente responsável (preencher o nome completo) e respectivas horas de contacto na unidade curricular:

Carlos Augusto Isaac Piló Viegas Damásio (T-28h; P-28h)

3.3.3. Outros docentes que leccionam a unidade curricular e respectivas horas de contacto na unidade curricular:

N/A

3.3.4. Objectivos de aprendizagem da unidade curricular (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber:

- *Linguagem para especificação de informação sobre recursos na Web (RDF)*
- *Princípios da “Linked Open Data”*
- *Linguagem de consulta SPARQL*
- *Linguagem para representação de ontologias (OWL)*
- *Linguagem de regras para a Web Semântica*
- *Limitações e complexidade das linguagens*
- *Diferença entre raciocínio em mundo aberto e mundo fechado*

Saber Fazer:

- *Desenvolver uma pequena ontologia em RDF(S) e OWL*
- *Utilizar um triplestore e interrogá-lo em SPARQL*
- *Combinar uma triplestore com dados abertos disponibilizados por um SPARQL endpoint disponível na Web Semântica*
- *Utilizar uma ferramenta de edição de ontologias*
- *Utilizar um motor de inferência para deduzir conhecimento na Web Semântica*

Competências Complementares:

- *Explorar autonomamente a literatura atualizada de um tópico*
- *Desenvolver o espírito crítico relativamente a tecnologia recente*
- *Trabalhar em equipa*
- *Efetuar uma apresentação oral expondo um tema recente*
- *Avaliar um trabalho científico*

3.3.4. Intended learning outcomes of the curricular unit (knowledge, skills and competences to be developed by the students):

Knowledge:

- *Language for specifying information about Web resources (RDF)*
- *Linked open data principles*
- *SPARQL query language*
- *Language for expressing ontologies (OWL)*
- *Rule languages for the Semantic Web*
- *Limitations and complexity of the reasoning tasks*
- *Distinction between open and closed world reasoning*

Application:

- *To develop a small ontology in RDF(S) and OWL*
- *Use a triplestore and query it in SPARQL*
- *Combine a triplestore with a linked open data SPARQL endpoint in the Semantic Web*

- *To use an ontology editor tool*
- *To use an inference engine to deduce knowledge from the Semantic Web*

Soft-skills:

- *To explore autonomously the recent bibliography of a topic*
- *To develop critical reasoning regarding recent technology*
- *To work in a team*
- *To orally present a survey on a recent topic*
- *To review a scientific work*

3.3.5. Conteúdos programáticos:

Teóricas (24h aulas+4h testes):

(2h)O que é a Semantic Web?

(2h)UNICODE, URIs e IRIs, XML Base, XML Namespaces, e XML Canonicalization

(8h)A linguagem e semântica do RDF(S) para representar conhecimento na Web Semântica. Teoria de modelos e os mecanismos de inferência da RDF(S), assim como as suas aplicações práticas e limitações

(5h)Estudo da linguagem SPARQL quer a nível de sintaxe, semântica e tradução para operadores de álgebra relacional e respetivo mapeamento em bases de dados. Tratamento de grandes volumes de dados

(5h)Lógicas terminológicas como um melhor formalismo de representação do conhecimento. As suas construções e semânticas são introduzidas, assim como os correspondentes algoritmos para fragmento ALCQ. A linguagem e semântica da OWL são apresentadas e aplicações práticas são ilustradas

(2h)Integração de ontologias com regras na Web Semântica: linguagens SWRL e RIF

Práticas: (8h) Jena, (4h) SPARQL, (4h) Protégé, (4h) Pellet reasoner, (4h) exercícios, (4h) projeto

3.3.5. Syllabus:

Theory (24h classes+4h midterms):

(2h)What is the Semantic Web?

(2h)Basics: UNICODE, URIs and IRIs, XML Base, XML Namespaces, and XML Canonicalization

(8h)The language and semantics of RDF(S) for representing knowledge in the Semantic Web. RDF(S) model theory and inference mechanisms are also addressed, as well as practical applications and its limitations

(5h)Study of the SPARQL language at the syntactic and semantic levels. Translation into relational algebra and mapping into databases. Challenges on dealing with large volumes of data

(5h)Description Logics are then introduced as a better knowledge representation formalism. Its constructs and semantics are introduced, as well the basic reasoning tasks and corresponding algorithms for the ALCQ fragment. The OWL language is presented and applications are provided

(2h)Integration of ontologies and rules in the Semantic Web: SWRL and RIF

Labs: (8h) Jena, (4h) SPARQL,(4h) Protégé,(4h) Pellet reasoner,(4h) exercises, (4h) project

3.3.6. Demonstração da coerência dos conteúdos programáticos com os objectivos de aprendizagem da unidade curricular:

A UC apresenta uma panorâmica atualizada da Web Semântica, seguindo uma estrutura lógica das matérias de acordo com os manuais de referência, abordando os “saberes” nas aulas teóricas. Trata-se o Resource Description Framework (RDF) e o vocabulário RDF Schema para expressar conhecimento na Web Semântica. São estudados os mecanismos de inferência da linguagem e sua complexidade, assim como a aplicação a casos concretos. A utilização da linguagem SPARQL para interrogação de informação na Web em RDF Schema será detalhada quer a nível abstrato quer a nível prático. Para lidar com conhecimento ontológico, introduz-se a linguagem OWL como complemento às capacidades expressivas do RDF Schema. Conclui-se a unidade curricular com a camada de regras da Web Semântica. Os “saber-fazer” são obtidos nas aulas práticas onde são exploradas autonomamente aplicações e ferramentas na resolução de tarefas e problemas dados. As competências são adquiridos com a realização do trabalho prático.

3.3.6. Demonstration of the syllabus coherence with the curricular unit’s intended learning outcomes:

This course presents an up-to-date UC view of the Web Semantic, following a logical structure according to the reference manuals, addressing the knowledge learning outcomes in the lectures. It is presented the Resource Description Framework (RDF) and RDF Schema vocabulary to express knowledge in the Semantic Web. The inference mechanisms of the language and their complexity, and are applied to concrete reasoning tasks. The use of the SPARQL language to query information expressed in RDF and RDFS is detailed at abstract and practical levels. In order to deal with ontological knowledge, the OWL language is studied complementing the RDFS expressiveness. The course concludes with the presentation of the rule’s layer of the Semantic Web. The “application” objectives are obtained in the lab classes where students explore tools and applications for solving given specific tasks and problems. The competences are acquired with the practical project work.

3.3.7. Metodologias de ensino (avaliação incluída):

Esta unidade curricular tem um nível avançado e visa ter uma componente de introdução à investigação. Nas aulas teóricas (2 horas semanais) são expostos os tópicos centrais que introduzem o aluno aos conceitos principais da Web Semântica que lhe permitirão depois aprofundarem algum tema em particular. Os alunos deverão entregar um trabalho prático sobre um tema, à sua escolha, relativo à Web Semântica. Os alunos deverão autonomamente explorar a informação existente na literatura, sendo a bibliografia inicial fornecida pelo docente.

As aulas práticas (2 horas semanais) servem para os alunos numa primeira fase explorarem ferramentas da Web Semântica existentes e posteriormente terem acompanhamento na realização do seu projeto.

A avaliação da unidade curricular é constituída por 2 testes individuais (cada 25%), um projeto/trabalho final de grupo (35% da nota), apresentação oral em Inglês e discussão de um trabalho de um colega (15% da nota).

A unidade curricular é ministrada em Inglês.

3.3.7. Teaching methodologies (including evaluation):

This course has a component of introduction to research. In the lectures (2 hours per week) some current topics are selected and exposed, introducing the students to the main Semantic Web concepts and which will allow them to explore a particular topic. The students should deliver a final project/essay about a Semantic Web topic suggested by the student. The students are expected to autonomously search the relevant literature, starting from the bibliography given by the teacher.

The course has lab classes (2 hours weekly), where students will explore on their own existing Semantic Web tools, and afterwards have support during the development of their project.

The evaluation of the course on Semantic Web is formed by two written short midterms (25% of the grade each), a final project/essay (35%), a presentation in English of the project and discussion of a colleague's work (15%).

This course is taught in English.

3.3.8. Demonstração da coerência das metodologias de ensino com os objectivos de aprendizagem da unidade curricular:

Sendo uma área de conhecimento muito recente e em constante evolução, as metodologias de ensino estão concebidas para dotar o aluno com os fundamentos teóricos e práticos da Web Semântica, estimulando a sua autonomia e espírito crítico. Os tópicos elencados nos “saberes” são tratados nas aulas teóricas da unidade curricular, sequenciada de acordo com a estrutura lógica dos manuais adotados e sendo disponibilizados acetatos. Todos os módulos da unidade curricular apontam para leituras complementares de artigos científicos disponibilizados na página Web da cadeira ou disponíveis na Web, assim como apontadores para as recomendações do W3C. É dada especial ênfase aos algoritmos de inferência, capacidades expressivas de cada uma das linguagens, limitações e complexidade temporal para estimular a discussão e identificar claramente os compromissos.

Os objetivos da componente “Saber-fazer” são garantidos com as aulas práticas, onde os alunos contactam com ferramentas e metodologias disponíveis, cujos algoritmos de base foram estudados nas aulas teóricas. As ferramentas são instaladas pelos alunos em diversos sistemas de operação. As ferramentas são utilizadas na resolução de tarefas e problemas desenhados pelo docente de forma a ilustrar os aspetos mais salientes e distintivos, assim como dificuldades que podem encontrar na construção de aplicações reais. As aulas práticas são também utilizadas para realizar exercícios preparatórios para os testes.

Os testes realizados durante o semestre garantem que os alunos seguem a matéria e se encontram preparados para a realização do projeto/trabalho final. O projeto/trabalho final é uma componente essencial da avaliação possibilitando a aquisição das “Competências” identificadas nos objetivos da unidade curricular. O trabalho é realizado em grupos de 2 ou 3 elementos, versando um tópico proposto e identificado pelos alunos a partir da literatura científica da área e que normalmente aprofunda alguma parte da matéria teórica complementando com uma aplicação prática. Neste processo são acompanhados pelo docente, discutindo os tópicos possíveis e indicando bibliografia de base para o trabalho. Esta discussão do tema de trabalho é realizada parcialmente nas aulas práticas e durante o horário de atendimento do docente. O trabalho é redigido preferencialmente em língua inglesa e apresentado em inglês num “workshop” da unidade curricular no final do semestre. Adicionalmente, os alunos avaliam anonimamente um projeto dos outros colegas de acordo com formulários de revisão utilizados em conferências científicas. Os formulários de avaliação dos discentes são também classificados e integrados no cômputo da nota final.

3.3.8. Demonstration of the teaching methodologies coherence with the curricular unit's intended learning outcomes:

Being a recent area of computer science and in fast evolution, the teaching methodologies are conceived to provide the accepted theoretical and practical fundamentals of the Semantic Web but at the same time stimulate autonomy and critical spirit. The “Knowledge” topics are addressed in the lectures, according to the logical sequence of the reference manuals and slides are made available. All the modules of the course provide pointers to scientific papers with complementary readings, and the W3C recommendations. We focus in the inference algorithms, expressive capabilities of the languages, limitations and computational complexity in order

to stimulate discussion and clearly identifying the compromises.

The “application” outcomes are attained in the lab classes, where the students use the tools and methodologies of the Semantic Web, whose algorithms have been studied in the lectures. The tools are installed by the students in different operating systems, and are used to solved tasks and problems designed by the lecturer to illustrate the most important and distinct issues, as well as the difficulties that can be found in the development of real applications. The lab classes are also used to solve preparatory exercises for the midterms, e.g. for understanding the major algorithms.

The midterms guarantee that the students do study at the appropriate time the subjects in order to be able to develop their final project. The final project is an essential component of the evaluation allowing for the achievement of the competences learning outcomes. The project is developed by teams of 2 or 3 students about a topic proposed and identified by them. The topic is proposed and identified by the students by analysing the scientific literature of the area, and that usually expands part of the subjects studied in the classes complemented with some application or practical work. In this process, the students are accompanied by the lecturer by discussing the topics and helping them in the identification of the basic literature. The discussion of the topic is partially performed in the lab classes and in the office hours. The work is preferably written in English and presented in a course’s “workshop” at the end of the semester. Additionally, the students review other colleagues’ projects using typical conference review forms. The reviews are classified by the lecturer and used to obtain part of the final grade.

3.3.9. Bibliografia principal:

Livros recomendados/adopted text books:

- *A Semantic Web Primer, 2nd Edition*

Grigoris Antoniou and Frank van Harmelen.

MIT Press, março 2008.

ISBN: 0-262-01242-1

- *Foundations of Semantic Web Technologies*

Pascal Hitzler, Markus Krotzsch and Sebastian Rudolph.

Chapman & Hall/CRC, 2009.

ISBN: 9781420090505.

- *The Description Logic Handbook. Theory, Implementation and Applications [Paperback]*

Edited by Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi and Peter Patel-Schneider.

Cambridge University Press, 2010, 2nd Edition.

ISBN: 0521150116

Os livro de texto são complementado com diversos artigos disponíveis na Web, em particular recomendações do W3C e outras entidades normativas. Essa bibliografia é acessível a partir da página da cadeira.

(Additional online bibliography is available from the course's web page, in particular W3C recommendations.)

4. Descrição e fundamentação dos recursos docentes

4.1 Descrição e fundamentação dos recursos docentes

4.1.1. Fichas curriculares dos docentes

Mapa V - Adriano Martins Lopes

4.1.1.1. Nome do docente (preencher o nome completo):

Adriano Martins Lopes

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Ana Maria Dinis Moreira

4.1.1.1. Nome do docente (preencher o nome completo):

Ana Maria Dinis Moreira

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - António Maria Lobo César Alarcão Ravara

4.1.1.1. Nome do docente (preencher o nome completo):

António Maria Lobo César Alarcão Ravara

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Artur Miguel de Andrade Vieira Dias

4.1.1.1. Nome do docente (preencher o nome completo):

Artur Miguel de Andrade Vieira Dias

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Carla Maria Gonçalves Ferreira

4.1.1.1. Nome do docente (preencher o nome completo):

Carla Maria Gonçalves Ferreira

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Carlos Augusto Isaac Piló Viegas Damásio

4.1.1.1. Nome do docente (preencher o nome completo):

Carlos Augusto Isaac Piló Viegas Damásio

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Carmen Pires Morgado

4.1.1.1. Nome do docente (preencher o nome completo):

Carmen Pires Morgado

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Fernanda Maria Barquinha Tavares Vieira Barbosa

4.1.1.1. Nome do docente (preencher o nome completo):

Fernanda Maria Barquinha Tavares Vieira Barbosa

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Fernando Pedro Reino da Silva Birra

4.1.1.1. Nome do docente (preencher o nome completo):

Fernando Pedro Reino da Silva Birra

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Francisco de Moura e Castro Ascensão de Azevedo

4.1.1.1. Nome do docente (preencher o nome completo):

Francisco de Moura e Castro Ascensão de Azevedo

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Francisco José Massa de Menezes

4.1.1.1. Nome do docente (preencher o nome completo):

Francisco José Massa de Menezes

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar convidado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

50

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Henrique João Lopes Domingos

4.1.1.1. Nome do docente (preencher o nome completo):

Henrique João Lopes Domingos

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Hervé Miguel Cordeiro Paulino

4.1.1.1. Nome do docente (preencher o nome completo):

Hervé Miguel Cordeiro Paulino

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - João Alexandre Carvalho Pinheiro Leite

4.1.1.1. Nome do docente (preencher o nome completo):

João Alexandre Carvalho Pinheiro Leite

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - João Baptista da Silva Araújo Junior

4.1.1.1. Nome do docente (preencher o nome completo):

João Baptista da Silva Araújo Junior

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - João Carlos Gomes Moura Pires

4.1.1.1. Nome do docente (preencher o nome completo):

João Carlos Gomes Moura Pires

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - João Manuel dos Santos Lourenço

4.1.1.1. Nome do docente (preencher o nome completo):

João Manuel dos Santos Lourenço

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - João Miguel da Costa Magalhães

4.1.1.1. Nome do docente (preencher o nome completo):

João Miguel da Costa Magalhães

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - João Ricardo Viegas da Costa Seco

4.1.1.1. Nome do docente (preencher o nome completo):

João Ricardo Viegas da Costa Seco

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Joaquim Francisco Ferreira da Silva

4.1.1.1. Nome do docente (preencher o nome completo):

Joaquim Francisco Ferreira da Silva

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Jorge Carlos Ferreira Rodrigues da Cruz

4.1.1.1. Nome do docente (preencher o nome completo):

Jorge Carlos Ferreira Rodrigues da Cruz

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - José Alberto Cardoso e Cunha

4.1.1.1. Nome do docente (preencher o nome completo):

José Alberto Cardoso e Cunha

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Catedrático ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - José Augusto Legatheaux Martins

4.1.1.1. Nome do docente (preencher o nome completo):

José Augusto Legatheaux Martins

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Catedrático ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - José Júlio Alves Alferes

4.1.1.1. Nome do docente (preencher o nome completo):

José Júlio Alves Alferes

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Catedrático ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Ludwig Krippahl

4.1.1.1. Nome do docente (preencher o nome completo):

Ludwig Krippahl

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Luís Manuel Marques da Costa Caires

4.1.1.1. Nome do docente (preencher o nome completo):

Luís Manuel Marques da Costa Caires

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Catedrático ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Manuel João Toscano Próspero dos Santos

4.1.1.1. Nome do docente (preencher o nome completo):

Manuel João Toscano Próspero dos Santos

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Margarida Paula Neves Mamede

4.1.1.1. Nome do docente (preencher o nome completo):

Margarida Paula Neves Mamede

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Maria Armanda Simenta Rodrigues Grueau

4.1.1.1. Nome do docente (preencher o nome completo):

Maria Armanda Simenta Rodrigues Grueau

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Maria Cecília Farias Lorga Gomes

4.1.1.1. Nome do docente (preencher o nome completo):

Maria Cecília Farias Lorga Gomes

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Miguel Carlos Pacheco Afonso Goulão

4.1.1.1. Nome do docente (preencher o nome completo):

Miguel Carlos Pacheco Afonso Goulão

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Miguel Jorge Tavares Pessoa Monteiro

4.1.1.1. Nome do docente (preencher o nome completo):

Miguel Jorge Tavares Pessoa Monteiro

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Nuno Manuel Ribeiro Preguiça

4.1.1.1. Nome do docente (preencher o nome completo):

Nuno Manuel Ribeiro Preguiça

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Nuno Manuel Robalo Correia

4.1.1.1. Nome do docente (preencher o nome completo):

Nuno Manuel Robalo Correia

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Catedrático ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Nuno Miguel Cavalheiro Marques

4.1.1.1. Nome do docente (preencher o nome completo):

Nuno Miguel Cavalheiro Marques

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Paulo Orlando Reis Afonso Lopes

4.1.1.1. Nome do docente (preencher o nome completo):

Paulo Orlando Reis Afonso Lopes

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Pedro Abílio Duarte de Medeiros

4.1.1.1. Nome do docente (preencher o nome completo):

Pedro Abílio Duarte de Medeiros

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Pedro Manuel Corrêa Calvente Barahona

4.1.1.1. Nome do docente (preencher o nome completo):

Pedro Manuel Corrêa Calvente Barahona

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Catedrático ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Rodrigo Seromenho Miragaia Rodrigues

4.1.1.1. Nome do docente (preencher o nome completo):

Rodrigo Seromenho Miragaia Rodrigues

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Sérgio Marco Duarte

4.1.1.1. Nome do docente (preencher o nome completo):

Sérgio Marco Duarte

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Sofia Carmen Faria Maia Cavaco

4.1.1.1. Nome do docente (preencher o nome completo):

Sofia Carmen Faria Maia Cavaco

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Susana Maria Santos Nascimento Martins Almeida

4.1.1.1. Nome do docente (preencher o nome completo):

Susana Maria Santos Nascimento Martins Almeida

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Teresa Isabel Lopes Romão

4.1.1.1. Nome do docente (preencher o nome completo):

Teresa Isabel Lopes Romão

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Vasco Miguel Moreira do Amaral

4.1.1.1. Nome do docente (preencher o nome completo):

Vasco Miguel Moreira do Amaral

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Vítor Manuel Alves Duarte

4.1.1.1. Nome do docente (preencher o nome completo):

Vítor Manuel Alves Duarte

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Ana Luísa da Graça Batista Custódio

4.1.1.1. Nome do docente (preencher o nome completo):

Ana Luísa da Graça Batista Custódio

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - João Filipe Lita da Silva

4.1.1.1. Nome do docente (preencher o nome completo):

João Filipe Lita da Silva

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Pamela Pacciani

4.1.1.1. Nome do docente (preencher o nome completo):

Pamela Pacciani

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Assistente ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

75

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - António José Mesquita da Cunha Machado Malheiro

4.1.1.1. Nome do docente (preencher o nome completo):

António José Mesquita da Cunha Machado Malheiro

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Maria de Fátima Vale de Gato Santos Rodrigues

4.1.1.1. Nome do docente (preencher o nome completo):

Maria de Fátima Vale de Gato Santos Rodrigues

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Maria do Carmo Proença Caseiro Brás

4.1.1.1. Nome do docente (preencher o nome completo):

Maria do Carmo Proença Caseiro Brás

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Ruy Araújo da Costa

4.1.1.1. Nome do docente (preencher o nome completo):

Ruy Araújo da Costa

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Pedro José dos Santos Palhinhas Mota

4.1.1.1. Nome do docente (preencher o nome completo):

Pedro José dos Santos Palhinhas Mota

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Bento José Carrilho Miguens Louro

4.1.1.1. Nome do docente (preencher o nome completo):

Bento José Carrilho Miguens Louro

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Maria Fernanda Alves da Veiga de Oliveira

4.1.1.1. Nome do docente (preencher o nome completo):

Maria Fernanda Alves da Veiga de Oliveira

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Isabel Maria Oitavem Fonseca da Rocha Kahle

4.1.1.1. Nome do docente (preencher o nome completo):

Isabel Maria Oitavem Fonseca da Rocha Kahle

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Dora Susana Raposo Prata Gomes

4.1.1.1. Nome do docente (preencher o nome completo):

Dora Susana Raposo Prata Gomes

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Isabel Cristina Maciel Natário

4.1.1.1. Nome do docente (preencher o nome completo):

Isabel Cristina Maciel Natário

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Luis Filipe dos Santos Gomes

4.1.1.1. Nome do docente (preencher o nome completo):

Luis Filipe dos Santos Gomes

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Pedro Miguel Negrão Maló

4.1.1.1. Nome do docente (preencher o nome completo):

Pedro Miguel Negrão Maló

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Assistente ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Anikó Katalin Horvath da Costa

4.1.1.1. Nome do docente (preencher o nome completo):

Anikó Katalin Horvath da Costa

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - José Carlos Ribeiro Kullberg

4.1.1.1. Nome do docente (preencher o nome completo):

José Carlos Ribeiro Kullberg

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - José Manuel Matos Ribeiro da Fonseca

4.1.1.1. Nome do docente (preencher o nome completo):

José Manuel Matos Ribeiro da Fonseca

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Nelson Fernando Chibeles Pereira Martins

4.1.1.1. Nome do docente (preencher o nome completo):

Nelson Fernando Chibeles Pereira Martins

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Maria Paula Pires dos Santos Diogo

4.1.1.1. Nome do docente (preencher o nome completo):

Maria Paula Pires dos Santos Diogo

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Catedrático ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Ana Maria de Oliveira Carneiro

4.1.1.1. Nome do docente (preencher o nome completo):

Ana Maria de Oliveira Carneiro

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - José Luís Câmara Toivola Leme

4.1.1.1. Nome do docente (preencher o nome completo):

José Luís Câmara Toivola Leme

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Bruno Rodrigues Vaz

4.1.1.1. Nome do docente (preencher o nome completo):

Bruno Rodrigues Vaz

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Monitor ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

30

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Ana Maria Manteigas Pedro

4.1.1.1. Nome do docente (preencher o nome completo):

Ana Maria Manteigas Pedro

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Vitor Hugo Bento Dias Fernandes

4.1.1.1. Nome do docente (preencher o nome completo):

Vitor Hugo Bento Dias Fernandes

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Carlos Manuel Agra Coelho

4.1.1.1. Nome do docente (preencher o nome completo):

Carlos Manuel Agra Coelho

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Associado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Pedro Alexandre da Rosa Corte Real

4.1.1.1. Nome do docente (preencher o nome completo):

Pedro Alexandre da Rosa Corte Real

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Jorge Alexandre Monteiro Carvalho Silva

4.1.1.1. Nome do docente (preencher o nome completo):

Jorge Alexandre Monteiro Carvalho Silva

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - José Luís Constantino Ferreira

4.1.1.1. Nome do docente (preencher o nome completo):

José Luís Constantino Ferreira

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - André João Maurício Leitão do Valle Wemans

4.1.1.1. Nome do docente (preencher o nome completo):

André João Maurício Leitão do Valle Wemans

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Ana Sofia Dinis Esteves

4.1.1.1. Nome do docente (preencher o nome completo):

Ana Sofia Dinis Esteves

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Assistente ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

20

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Fernanda Antonia Josefa Llussá

4.1.1.1. Nome do docente (preencher o nome completo):

Fernanda Antonia Josefa Llussá

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Maria do Rosário de Meireles Ferreira Cabrita

4.1.1.1. Nome do docente (preencher o nome completo):

Maria do Rosário de Meireles Ferreira Cabrita

4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:

Professor Auxiliar convidado ou equivalente

4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):

100

4.1.1.6. Ficha curricular de docente:

[Mostrar dados da Ficha Curricular](#)

Mapa V - Rogério Salema Araújo Puga Leal**4.1.1.1. Nome do docente (preencher o nome completo):***Rogério Salema Araújo Puga Leal***4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):**

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:*Professor Auxiliar ou equivalente***4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):**

100

4.1.1.6. Ficha curricular de docente:[Mostrar dados da Ficha Curricular](#)**Mapa V - Virgílio António da Cruz Machado****4.1.1.1. Nome do docente (preencher o nome completo):***Virgílio António da Cruz Machado***4.1.1.2. Instituição de ensino superior (preencher apenas quando diferente da Instituição proponente mencionada em A1):**

<sem resposta>

4.1.1.3 Unidade Orgânica (preencher apenas quando diferente da unidade orgânica mencionada em A2):

<sem resposta>

4.1.1.4. Categoria:*Professor Catedrático ou equivalente***4.1.1.5. Regime de tempo na Instituição que submete a proposta (%):**

100

4.1.1.6. Ficha curricular de docente:[Mostrar dados da Ficha Curricular](#)**4.1.2 Equipa docente do ciclo de estudos****4.1.2. Equipa docente do ciclo de estudos / Study cycle's academic staff**

Nome / Name	Grau / Degree	Área científica / Scientific Area	Regime de tempo / Employment link	Informação/ Information
Adriano Martins Lopes	Doutor	Informática	100	Ficha submetida
Ana Maria Dinis Moreira	Doutor	Informática	100	Ficha submetida
António Maria Lobo César Alarcão Ravara	Doutor	Matemática	100	Ficha submetida
Artur Miguel de Andrade Vieira Dias	Doutor	Informática	100	Ficha submetida
Carla Maria Gonçalves Ferreira	Doutor	Computer Science	100	Ficha submetida
Carlos Augusto Isaac Piló Viegas Damásio	Doutor	Informática	100	Ficha submetida

Carmen Pires Morgado	Doutor	Informática	100	Ficha submetida
Fernanda Maria Barquinha Tavares Vieira Barbosa	Doutor	Informática	100	Ficha submetida
Fernando Pedro Reino da Silva Birra	Doutor	Informática	100	Ficha submetida
Francisco de Moura e Castro Ascensão de Azevedo	Doutor	Informática	100	Ficha submetida
Francisco José Massa de Menezes	Licenciado	Informática	50	Ficha submetida
Henrique João Lopes Domingos	Doutor	Informática	100	Ficha submetida
Hervé Miguel Cordeiro Paulino	Doutor	Informática	100	Ficha submetida
João Alexandre Carvalho Pinheiro Leite	Doutor	Informática	100	Ficha submetida
João Baptista da Silva Araújo Junior	Doutor	Computer Science	100	Ficha submetida
João Carlos Gomes Moura Pires	Doutor	Informática / Inteligência Artificial	100	Ficha submetida
João Manuel dos Santos Lourenço	Doutor	Informática	100	Ficha submetida
João Miguel da Costa Magalhães	Doutor	Computer Science	100	Ficha submetida
João Ricardo Viegas da Costa Seco	Doutor	Informática	100	Ficha submetida
Joaquim Francisco Ferreira da Silva	Doutor	Informática	100	Ficha submetida
Jorge Carlos Ferreira Rodrigues da Cruz	Doutor	Informática	100	Ficha submetida
José Alberto Cardoso e Cunha	Doutor	Informática	100	Ficha submetida
José Augusto Legatheaux Martins	Doutor	Informática - Sistemas Distribuídos	100	Ficha submetida
José Júlio Alves Alferes	Doutor	Informática	100	Ficha submetida
Ludwig Krippahl	Doutor	Bioquímica estrutural	100	Ficha submetida
Luís Manuel Marques da Costa Caires	Doutor	Informática	100	Ficha submetida
Manuel João Toscano Próspero dos Santos	Doutor	Informática	100	Ficha submetida
Margarida Paula Neves Mamede	Doutor	Informática	100	Ficha submetida
Maria Armanda Simenta Rodrigues Grueau	Doutor	Ciência de Informação	100	Ficha submetida
Maria Cecília Farias Lorga Gomes	Doutor	Informática	100	Ficha submetida
Miguel Carlos Pacheco Afonso Goulão	Doutor	Informática	100	Ficha submetida
Miguel Jorge Tavares Pessoa Monteiro	Doutor	Informática	100	Ficha submetida
Nuno Manuel Ribeiro Preguiça	Doutor	Informática	100	Ficha submetida
Nuno Manuel Robalo Correia	Doutor	Engenharia Informática e Electrotécnica	100	Ficha submetida
Nuno Miguel Cavalheiro Marques	Doutor	Informática	100	Ficha submetida
Paulo Orlando Reis Afonso Lopes	Doutor	Informática	100	Ficha submetida
Pedro Abílio Duarte de Medeiros	Doutor	Informática	100	Ficha submetida
Pedro Manuel Corrêa Calvente Barahona	Doutor	Computer Science	100	Ficha submetida
Rodrigo Seromenho Miragaia Rodrigues	Doutor	Informática	100	Ficha submetida
Sérgio Marco Duarte	Doutor	Informática	100	Ficha submetida
Sofia Carmen Faria Maia Cavaco	Doutor	Computer science	100	Ficha submetida
Susana Maria Santos Nascimento Martins Almeida	Doutor	Informática	100	Ficha submetida
Teresa Isabel Lopes Romão	Doutor	Engenharia do Ambiente	100	Ficha submetida
Vasco Miguel Moreira do Amaral	Doutor	Computer Science	100	Ficha submetida
Vítor Manuel Alves Duarte	Doutor	Informática	100	Ficha submetida
Ana Luísa da Graça Batista	Doutor	Matemática	100	Ficha submetida

Custódio					
João Filipe Lita da Silva	Doutor	Matemática, especialidade de Estatística	100		Ficha submetida
Pamela Pacciani	Licenciado	Matemática	75		Ficha submetida
António José Mesquita da Cunha Machado Malheiro	Doutor	Matemática	100		Ficha submetida
Maria de Fátima Vale de Gato Santos Rodrigues	Doutor	Matemática	100		Ficha submetida
Maria do Carmo Proença Caseiro Brás	Doutor	Matemática na especialidade de Investigação Operacional	100		Ficha submetida
Ruy Araújo da Costa	Doutor	Engenharia de Sistemas	100		Ficha submetida
Pedro José dos Santos Palhinhas Mota	Doutor	Matemática- Especialidade Estatística	100		Ficha submetida
Bento José Carrilho Miguens Louro	Doutor	Matemática	100		Ficha submetida
Maria Fernanda Alves da Veiga de Oliveira	Doutor	Matemática	100		Ficha submetida
Isabel Maria Oitavem Fonseca da Rocha Kahle	Doutor	Matemática	100		Ficha submetida
Dora Susana Raposo Prata Gomes	Doutor	Estatística	100		Ficha submetida
Isabel Cristina Maciel Natário	Doutor	Probabilidades e Estatística	100		Ficha submetida
Luis Filipe dos Santos Gomes	Doutor	Engenharia Electrotécnica - Sistemas Digitais	100		Ficha submetida
Pedro Miguel Negrão Maló	Mestre	Engenharia Informática	100		Ficha submetida
Anikó Katalin Horvath da Costa	Doutor	Engenharia Electrotécnica, especialidade de Sistemas Digitais	100		Ficha submetida
José Carlos Ribeiro Kullberg	Doutor	Geologia / Geologia Estrutural	100		Ficha submetida
José Manuel Matos Ribeiro da Fonseca	Doutor	Engenharia Electrotécnica	100		Ficha submetida
Nelson Fernando Chibeles Pereira Martins	Doutor	Engenharia de Sistemas	100		Ficha submetida
Maria Paula Pires dos Santos Diogo	Doutor	História da Ciência e da Tecnologia- Epistemologia das Ciências	100		Ficha submetida
Ana Maria de Oliveira Carneiro	Doutor	History, Philosophy and Social Relations of Science	100		Ficha submetida
José Luís Câmara Toivola Leme	Doutor	Epistemologia das ciências	100		Ficha submetida
Bruno Rodrigues Vaz	Licenciado	Matemática	30		Ficha submetida
Ana Maria Manteigas Pedro	Doutor	Equações Diferenciais Funcionais	100		Ficha submetida
Vitor Hugo Bento Dias Fernandes	Doutor	Matemática	100		Ficha submetida
Carlos Manuel Agra Coelho	Doutor	Estatística/Bioestatística	100		Ficha submetida
Pedro Alexandre da Rosa Corte Real	Doutor	Estatística	100		Ficha submetida
Jorge Alexandre Monteiro Carvalho Silva	Doutor	Física de Superfícies	100		Ficha submetida
José Luís Constantino Ferreira	Doutor	Física - Ramo Biofísica	100		Ficha submetida
André João Maurício Leitão do Valle Wemans	Doutor	Engenharia Física	100		Ficha submetida
Ana Sofia Dinis Esteves	Mestre	Biotechnology	20		Ficha submetida
Fernanda Antonia Josefa Llussa	Doutor	Economia	100		Ficha submetida
Maria do Rosário de Meireles Ferreira Cabrita	Doutor	Gestão	100		Ficha submetida
Rogério Salema Araújo Puga Leal	Doutor	Engenharia Industrial - Sistemas de Gestão	100		Ficha submetida
Virgílio António da Cruz Machado	Doutor	Computer Integrated Manufacturing	100		Ficha submetida

<sem resposta>

4.2. Dados percentuais da equipa docente do ciclo de estudos

4.2.1.a Número dos docentes do ciclo de estudos em tempo integral na Instituição:

76

4.2.1.b Percentagem dos docentes do ciclo de estudos em tempo integral na Instituição (campo de preenchimento automático calculado após a submissão do formulário):

97,7

4.2.2.a Número dos docentes do ciclo de estudos em tempo integral com uma ligação à Instituição por um período superior a três anos:

75

4.2.2.b Percentagem dos docentes do ciclo de estudos em tempo integral com uma ligação à Instituição por um período superior a três anos (campo de preenchimento automático calculado após a submissão do formulário):

96,5

4.2.3.a Número dos docentes do ciclo de estudos em tempo integral com grau de doutor:

75

4.2.3.b Percentagem dos docentes do ciclo de estudos em tempo integral com grau de doutor (campo de preenchimento automático calculado após a submissão do formulário):

96,5

4.2.4.a Número (ETI) de docentes do ciclo de estudos inscritos em programas de doutoramento há mais de um ano:

1

4.2.4.b Percentagem dos docentes do ciclo de estudos inscritos em programas de doutoramento há mais de um ano (campo de preenchimento automático calculado após a submissão do formulário):

1,3

4.2.5.a Número (ETI) de docentes do ciclo de estudos não doutorados com grau de mestre (pré-Bolonha):

1

4.2.5.b Percentagem dos docentes do ciclo de estudos não doutorados com grau de mestre (pré-Bolonha) (campo automático calculado após a submissão do formulário):

1,3

4.3. Procedimento de avaliação do desempenho

4.3. Procedimento de avaliação do desempenho do pessoal docente e medidas para a sua permanente actualização:

Em 16 de Agosto de 2010 foi publicado em DR (2ª Série, nº 158) o Regulamento nº 684/2010 relativo à Avaliação do Desempenho e Alteração do Posicionamento Remuneratório dos docentes da UNL-Universidade Nova de Lisboa.

O regulamento tem por objeto o desempenho dos docentes da UNL, visando avaliá-lo em função do mérito e melhorar a sua qualidade.

A avaliação de desempenho abrange todos os docentes da UNL, tem em conta a especificidade de cada área disciplinar e considera todas as vertentes da respetiva atividade:

a) Docência (e.g., diversidade de disciplinas ensinadas; disponibilização de material pedagógico; orientação de Dissertações de Mestrado e de Teses de Doutoramento; participação em júris);

b) Investigação científica, desenvolvimento e inovação (e.g., coordenação e participação em projetos de investigação e direção de unidades de investigação; publicação de artigos e livros; comunicações em congressos científicos; participação em órgãos de revistas científicas; patentes; participação em comissões, organizações ou redes científicas);

- c) *Tarefas administrativas e de gestão académica;*
- d) *Extensão universitária, divulgação científica e prestação de serviços à comunidade (e.g., prémios e distinções académicas; relatórios no âmbito do estatuto da carreira docente; serviços prestados a outras entidades).*

As ponderações a considerar em cada vertente são as seguintes:

- a) *Docência — entre 20 % e 70 %;*
- b) *Investigação científica, desenvolvimento e inovação — entre 20 % e 70 %;*
- c) *Tarefas administrativas e de gestão académica — entre 10 % e 40 %;*
- d) *Atividades de extensão universitária, divulgação científica e prestação de serviços à comunidade — entre 5% e 40%*

A avaliação positiva é expressa numa escala de três posições (mínimo de 3 pontos e máximo de 9 pontos).

Compete ao Conselho Científico a condução do processo de avaliação de desempenho. Compete ao Conselho Pedagógico pronunciar -se na generalidade sobre o processo de avaliação de desempenho. Compete ao Reitor da UNL homologar os resultados da avaliação do desempenho.

A avaliação do desempenho é feita uma vez em cada triénio, sem prejuízo da monitorização anual, e releva para os seguintes efeitos:

- a) *Contratação por tempo indeterminado dos professores auxiliares;*
- b) *Renovação dos contratos a termo certo dos docentes não integrados na carreira;*
- c) *Alteração do posicionamento remuneratório.*

Os docentes que acumulem um mínimo 18 pontos nas avaliações de desempenho deverão ter uma alteração do posicionamento remuneratório. Os docentes com avaliação considerada insuficiente em dois triénios consecutivos poderão sofrer as consequências previstas no Estatuto Disciplinar dos Trabalhadores que exercem Funções Públicas.

A FCT elaborou o seu Regulamento em consonância com o da UNL, tendo definido métricas específicas para as áreas da Ciência e Engenharia. O Regulamento da FCT já foi aprovado e publicado no DR, 2ª Série, nº 193 de 4 de outubro 2012 (Despacho 13109/2012).

4.3. Academic staff performance evaluation procedures and measures for its permanent updating:

The rules for Performance Evaluation and Amendment of Position Remuneration of academic staff of UNL Universidade Nova de Lisboa were officially published in August 16, 2010 (Regulation 684/2010).

The regulation concerns the performance of the UNL academic staff in order to evaluate it based on merit and improve its quality.

The performance evaluation covers all UNL academic staff, takes into account the specifics of each subject area and considers all aspects of their business:

- a) *Teaching (e.g., diversity of subjects taught, availability of teaching materials, supervision of Master and PhD, Theses, participation in boards of academic juries);*
- b) *Scientific research, development and innovation (e.g., coordination and participation in research projects and coordination of research units, publication of scientific articles and books, conference papers, participation in bodies of scientific journals, patents, participation in scientific committees, organizations or networks);*
- c) *Administrative and academic management activities (e.g., participation in bodies of UNL and UNL academic units);*
- d) *Extension activities, scientific dissemination and services delivery to the community (e.g., academic honours and awards, reports in the status of the teaching profession, services provision to other entities).*

The weights assigned to the above dimensions are:

- a) *Teaching - between 20% and 70%;*
- b) *Scientific research, development and innovation - between 20% and 70%;*
- c) *Administrative and academic management activities- between 10% and 40%;*
- d) *Extension activities, scientific dissemination and services delivery to the community - between 5% and 40%.*

The positive evaluation is expressed on a scale of three positions (minimum of 3 points and a maximum of 9 points).

At the academic unit level, the Scientific Council conducts the performance evaluation process and the Pedagogical Council issues an overall appreciation of it. The UNL Rector approves the results of the performance evaluation.

Performance evaluation is carried out once every three years, subject to annual monitoring, and is relevant for the following purposes:

- a) *Contract of assistant professors for an indefinite period;*
- b) *Renewal of temporary contracts for teachers that are not integrated in the regular academic career;*
- c) *Change of salary position.*

The salary position of teachers who accumulate a minimum of 18 points in performance evaluation may be upgraded. Teachers with performance evaluation considered insufficient in two consecutive three-year periods may suffer the consequences outlined in the Disciplinary Statute of Civil Servants.

FCT has developed its regulations in accordance with UNL's rules, having defined specific evaluation metrics for the Science and Engineering areas. The FCT regulations were already approved and officially published on the 4th of October 2012 (DR, 2ª Série, 193, Despacho 13109/2012)

5. Descrição e fundamentação de outros recursos humanos e materiais

5.1. Pessoal não docente afecto ao do ciclo de estudos:

Participam em atividades de suporte à gestão do ciclos de estudos o seguinte pessoal não docente:

Maria Filipa Sardinha Chamorrinha Mira dos Reis - Coordenadora Técnica

Anabela Fonseca Nunes da Silva Duarte - Assistente Técnica

Paula Maria Gomes do Amaral Brás - Assistente Técnica

Sandra Rosa Rafael Rainha - Assistente Técnica

Apoio à coordenação e disseminação do curso, apoio aos docentes, atendimento dos estudantes.

Mário Alexandre Calvelas Ribeiro - Técnico de Informática - Grau 1 - Nível 1

Apoio à gestão dos laboratórios de ensino, da redes de computadores, dos serviços de impressão, e da manutenção de informação na web, incluindo atividades de ligação com a Divisão de Informática, que também disponibiliza recursos humanos sempre que necessário.

5.1. Non academic staff allocated to the study cycle:

The following non academic staff participate in management support activities related to the study cycle:

Maria Filipa Sardinha Chamorrinha Mira dos Reis – Administrative Coordinator

Anabela Fonseca Nunes da Silva Duarte – Administrative Assistant

Paula Maria Gomes do Amaral Brás - Administrative Assistant

Sandra Rosa Rafael Rainha - Administrative Assistant

Support to program coordination and dissemination activities, support to the teaching staff, assistance and administrative interface with the students.

Mário Alexandre Calvelas Ribeiro – IT technical engineer - Grade 1 - Nível 1

Technical management of teaching laboratories, computer networks, printing services, web site information maintenance and update, including liaison activities with the central “Divisão de Informática”, which also contributes with IT human resources whenever necessary.

5.2. Instalações físicas afectas e/ou utilizadas pelo ciclo de estudos (espaços lectivos, bibliotecas, laboratórios, salas de computadores, etc.):

38 Gabinetes docentes e investigadores

3 Salas de reuniões com datashow de acesso livre

1 Sala de reuniões departamental com datashow e videoconferência

1 Sala de seminários com datashow e videoconferência

7 Gabinetes e salas para pessoal administrativo e técnico

6 Salas servidores e de infraestrutura informática (área total 100m²)

8 Laboratórios de Investigação

10 Laboratórios de Ensino Informática com datashow incluído laboratório específicos de redes e projetos de alunos

6 Salas alunos de utilização livre com computador ou mesa de trabalho

34 Infraestrutura de controle de acessos suportada em cartões magnéticos

10 Câmaras de video-vigilância

Vários espaços comuns; Biblioteca central da FCT UNL, Cafetarias, etc.

5.2. Facilities allocated and/or used by the study cycle (teaching spaces, libraries, laboratories, computer rooms, etc.):

38 Offices for teaching and research staff

3 Free access meeting rooms with datashow

1 Departmental meeting room with datashow e videoconferencing

1 Seminar room with datashow e videoconferencing

7 Offices and rooms for technical and administrative staff

6 Server rooms and e IT infrastructure space

8 Research Labs

10 Teaching Labs for students work and classes with datashow including specialized laboratories for computer networks and student projects

6 Free access student rooms with computer or working table

34 Access control infrastructure based on magnetic cards

10 Surveillance video-cameras

Several common spaces; Central Library FCT UNL, Cafeterias, etc.

5.3. Indicação dos principais equipamentos e materiais afectos e/ou utilizados pelo ciclo de estudos (equipamentos didácticos e científicos, materiais e TICs):

18 Equipamento data show (fixo)

3 Equipamento data show (portátil)

190 Computadores (para alunos)

70 Computadores (docentes)

10 Computadores (serviços técnicos e administrativos)

19 Computadores (serviços centrais)

12 Computadores (CPUs) agregados em cluster (alunos)

110 Computadores (CPUs) agregados em cluster (investigação)

17 Infraestruturas de rede sem fios (APs)

34 Infraestruturas de rede com fios (switches)

2 Firewalls

4 Impressoras para docentes, serviços e pós-graduação

2 Fotocopiadora

2 Equipamento de video-conferência; 1 Sistema de som

2 Televisor plasma

15 Computador portáteis tipo tablet

60 Monitores TFT 22" (para ligação de portáteis)

5.3. Indication of the main equipments and materials allocated and/or used by the study cycle (didactic and scientific equipments and materials and ICTs):

18 DataShow equipment (fixed)

3 DataShow equipment (mobile)

190 Computer workstations (for students)

70 Computer workstations (teaching staff)

10 Computer workstations (technical and administrative staff)

19 Computer workstations (central services)

12 Servers (CPUs) in cluster configuration (students)

110 Servers (CPUs) in cluster configuration (research)

17 Wireless infrastructures (APs)

34 Wireless infrastructures (switches)

2 Firewalls

4 Printers

2 Photocopier machines

2 Video conferencing equipment; 1 Sound system

2 Plasma TV

15 Tablet computers

60 TFT 22" monitors (for laptops)

6. Actividades de formação e investigação

6.1. Centro(s) de investigação, na área do ciclo de estudos, em que os docentes desenvolvem a sua actividade científica

6.1. Mapa VI Centro(s) de investigação, na área do ciclo de estudos, em que os docentes desenvolvem a sua actividade científica / Research Centre(s) in the area of the study cycle, where the members of the academic staff develop their scientific activities

	Classificação		
Centro de Investigação /	(FCT) /	IES /	Observações / Observations
Research Centre	Classification	Institution	

(FCT)

CITI - Centro de Informática e Tecnologias da Informação/Research Center for Informatics and Information Technologies (http://citi.di.fct.unl.pt)	Bom/Good	FCT-UNL	A agenda de investigação do CITI (28 FTE) está alinhada com o tema “Princípios e Engenharia de Serviços de Software e Media” / The research agenda of CITI is currently focused on the development of new Principles and Engineering for Networked Software Services and Media
CENTRIA - Centro de Inteligência Artificial / Centre for Artificial Intelligence (http://centria.di.fct.unl.pt/)	Bom/Good	FCT-UNL	O CENTRIA (15 FTE) desenvolve investigação em Inteligência Artificial / CENTRIA develops research on Artificial Intelligence
-	-	-	Atividade dos centros refletida nos 159 artigos referidos em 6.2, em mais 693 artigos em conferências internacionais relevantes e nos projetos e outras atividades referidos em 6.3 / The main results of the Centres activity are the 159 articles mentioned in 6.2, more 693 articles published in highly relevant international conferences, and the projects and other activities mentioned in 6.3.

Perguntas 6.2 e 6.3

6.2. Indicação do número de publicações científicas do corpo docente do ciclo de estudos, na área predominante do ciclo de estudos, em revistas internacionais com revisão por pares nos últimos cinco anos:

159

6.3. Lista dos principais projectos e/ou parcerias nacionais e internacionais em que se integram as actividades científicas, tecnológicas, culturais e artísticas desenvolvidas na área de ciclo de estudos:

O DI e os seus centros têm estado sempre envolvidos em redes e projetos nacionais (FCT/MCT, QREN, ADI) da UE (ERC, Strep, IP, NoE, COST), e outros como os programas Carnegie-Mellon e UT Austin Portugal, o Mestrado Europeu em Lógica Computacional, e AsiaLink. O DI está envolvido em mais de 30 projetos financiados, alguns por empresas. Colaboramos regularmente com escolas nacionais (IST, U. Lisboa, U. Minho, U. Porto, U. Aveiro, U. Coimbra, U. Évora e outras). Alunos de doutoramento do DI são orientados com Carnegie-Mellon, UT Austin, IMT Lucca, Max Planck Institute, Wright U., Columbia U., UFRGS. Partilhamos UC com programas doutorais do IST, UL e vice versa (protocolo NICE). Decorrem colaborações com muitas universidades e empresas internacionais e nacionais (INRIA, Google, Microsoft Research, IBM, Imperial College, Dresden U, ESA, OutSystems, Zon, RTP, Novabase, Edisoft). A lista completa de projetos pode ser consultada no site web dos nossos centros de investigação.

6.3. Indication of the main projects and/or national and international partnerships where the scientific, technological, cultural and artistic activities developed in the area of the study cycle are integrated:

The Department and its research units have been involved in research networks and projects, through national (FCT/MCT, QREN, ADI), UE funded (ERC, Strep, IP, NoE, COST), the Carnegie Mellon and UT Austin partnerships, the European Master in Computational Logic, AsiaLink. We collaborate with national schools (IST, U. Lisboa, U. Minho, U. Porto, U. Aveiro, U. Coimbra, U. Évora and other). We are currently involved in more than 30 funded projects, some by companies. Several of our PhD students are co-supervised with institutions such as Carnegie Mellon, UT Austin, IMT Lucca, Max Planck Institute, Columbia University, Wright University, UFRGS. We share PhD courses with IST and UL (NICE framework). Many national and international academic and industrial collaborations are ongoing (INRIA, Google, Microsoft Research, IBM, Imperial College, Dresden U, ESA, OutSystems, Zon, RTP, Novabase, Edisoft). Complete list of projects and partnerships may be found in the web site of our research units.

7. Actividades de desenvolvimento tecnológico, prestação de serviços à comunidade e formação avançada

7.1. Descreva estas actividades e se a sua oferta corresponde às necessidades do mercado, à missão e aos objectivos da Instituição:

Várias atividades do Departamento e centros conduzem a transferência de tecnologia e de conhecimento, realizada muitas vezes em colaboração com empresas nacionais e internacionais, gerando inovação e fixando

propriedade intelectual, em alinhamento com a missão da FCTUNL. Endereçam múltiplos aspectos e necessidades, exemplos incluem a segurança de aplicações web, televisão imersiva, gestão de emergências, tecnologias multimedia em espaços públicos, monitorização dos oceanos, suporte ao desenvolvimento artístico, detecção de padrões de utilizadores, análise de redes de energia, entre outros. O curso proposto oferece aos estudantes oportunidades de atividade no exterior da escola, nomeadamente na UC “Actividade Prática de Desenvolvimento Curricular”, e também a possibilidade de realizarem a dissertação em ambiente académico-empresarial, tendo o DI e a FCTUNL estabelecido protocolos de enquadramento e parcerias com várias empresas (ver mais info no site web do DI).

7.1. Describe these activities and if they correspond to market needs and to the mission and objectives of the Institution:

Several activities of the Department and research units lead to technology and knowledge transfer, often developed in collaboration with national and international firms, producing innovation and creating intellectual property, in alignment with FCTUNL's mission. Activities address multiple aspects and needs, examples include web application security, immersive television, emergency management, multimedia technology in public spaces, ocean monitoring, artistic development support, user behavior pattern detection, energy grid analysis, to list a few. The program proposed offers to students the opportunity of developing activities in external placement, in particular in the CU “Practical Activity for Curricular Development”, and also the possibility of developing the dissertation in a collaboration between academy and a firm, for which the DI and FCTUNL have established agreements with several firms (more info on DI web site).

8. Enquadramento na rede de formação nacional da área (ensino superior público)

8.1. Avaliação da empregabilidade dos graduados por ciclos de estudos similares com base nos dados do MEE:

De acordo com os dados citados pelo inquérito estatístico anual sobre estudantes inscritos e diplomados (RAIDES), para estudantes diplomados entre 2000-2001 e 2009-2010, recolhidos do site da DGES, a empregabilidade para a área geral CNAEF “523 Electrónica e Automação” é de 98%, sendo que para os cursos de Engenharia Informática da FCT/UNL de 99% (o número de inscritos nos centros de emprego tem como fonte o Instituto de Emprego e Formação Profissional). De acordo com a Divisão Académica da FCT, com base em informação recolhida através de Questionário online divulgado em Outubro de 2009 aos diplomados pela FCT/UNL entre 2004 e Julho de 2009, a percentagem de diplomados em Engenharia Informática da FCT/UNL com emprego ao fim de um ano é de 99%. Desde 1999, o Departamento de Informática da FCT/UNL graduou mais de 1500 licenciados e mestres em Engenharia Informática.

8.1. Evaluation of the graduates' employability based on MEE data:

According to data collected from the annual statistical survey on graduated and enrolled students (RAIDES), and for students graduating between 2000-2001 and 2009-2010 (source DGES web site), the employment rate for the area CNAEF “523 Electrónica e Automação” is of 98%, and for the FCT UNL programs in Computer Science and Engineering of 99% (the number of graduates registered in employment centres was obtained from Instituto de Emprego e Formação Profissional). According to the FCT Academic Services, based in information collected through an online survey conducted in October 2009 to alumni that graduated between 2004 and July 2009, 99% of Computer Science and Engineering graduates from FCT find a job position after at most one year. Since 1999, the Departamento de Informática FCT/UNL has delivered more than 1500 “licenciados” and master level graduates in Computer Science and Engineering.

8.2. Avaliação da capacidade de atrair estudantes baseada nos dados de acesso (DGES):

A FCTUNL demonstrou ao longo de décadas a capacidade sustentada de atrair candidatos em qualidade e quantidade para a sua oferta educativa em Engenharia Informática. De acordo com a DGES, o número de candidatos que nos últimos anos manifestou interesse na Engenharia Informática da FCTUNL foi em média de cerca de 660 por ano, com nota de candidatura de mínimo 13. Tanto a Licenciatura pré-bolonha, como a Licenciatura de 1º ciclo preencheram normalmente todas as vagas disponíveis (~150 / ano). Em 2012/13 foram colocados 150 candidatos, esgotando-se toda a oferta na 1ª Fase. Os candidatos aos cursos de Informática da Nova obtêm em grande maioria a sua colocação em primeira opção. Por outro lado, a maioria destes posiciona-se no sentido de obter uma formação de dois ciclos em Engenharia Informática. Conjugando estes fatores com os altos níveis de empregabilidade dos cursos na área da Informática acredita-se que o Mestrado Integrado em Engenharia Informática apresenta um alto potencial de atração.

8.2. Evaluation of the capacity to attract students based on access data (DGES):

FCTUNL has demonstrated for decades a sustained ability to attract candidates for its educational offer in Computer Science and Engineering in both quality and quantity. According to DGES, 660 candidates per year on average have demonstrated interest in FCT UNL Computer Science program, with a application grade of 130. Both the pre bologna “licenciatura” (5-years) and the 1st cycle BSc program normally exhausted all available positions (~150 / year). For 2012/13 new 150 candidates were admitted, having the full offer being filled during the 1st Phase. Candidates to Nova’s Computer Science and Engineering program obtain their admission as their first option. On the other hand, the majority intends to target a complete 2-cycle education in Computer Science and Engineering. Putting all these factors together jointly with the high employment rate level of all programs in the area of Informatics, we believe that the integrated master now proposed presents an high attraction potential.

8.3. Lista de parcerias com outras Instituições da região que leccionam ciclos de estudos similares:

A nível da formação de 1º e 2º ciclo não estão estabelecidas parcerias formais com instituições da região. No entanto, o Departamento de Informática mantém colaborações estreitas com escolas congéneres na região como o Instituto Superior Técnico da Universidade Técnica de Lisboa e com a Faculdade de Ciências da Universidade de Lisboa, no contexto de avaliação de dissertações de mestrado, colaboração em projetos de investigação em que somos parceiros, e acarinha a manutenção de canais de comunicação e troca de experiências. Tais parcerias fazem particular sentido e poderão vir a ser melhor exploradas no contexto de 3º ciclos, onde já existem algumas iniciativas preliminares, como a iniciativa NICE, visando a colaboração entre os programas doutorais das escolas referidas.

8.3. List of partnerships with other Institutions in the region teaching similar study cycles:

Concerning the education at the 1st and 2nd cycle, we have no established partnerships with other local institutions. However, our Department is developing close collaboration with nearby engineering and science higher education schools such as the Instituto Superior Técnico da Universidade Técnica de Lisboa and the Faculdade de Ciências da Universidade de Lisboa, for example, in the evaluation of master dissertations, research project collaborations, and actively sustains and cherishes open communication and interchange channels. Such partnerships make particular sense and should be better explored in the context of doctoral programs, some preliminary initiatives are already in place, such as the NICE framework, aiming at developing collaboration between doctoral programs of the above mentioned schools.

9. Fundamentação do número total de ECTS do novo ciclo de estudos**9.1. Justificação do número total de unidades de crédito e da duração do ciclo de estudos com base no determinado nos artigos 8.º ou 9.º (1.º ciclo), 18.º (2.º ciclo), 19.º (mestrado integrado) e 31.º (3.º ciclo) do Decreto-Lei n.º 74/2006, de 24 de Março:**

De acordo com o Artigo 19º do Decreto-Lei no 74/2006, e tratando-se de um ciclo de estudos integrado, com 5 anos (10 semestres), foi atribuído ao ciclo de estudos um total de 300 ECTS para a obtenção do grau de Mestre. Ao trabalho de dissertação de mestrado é atribuído um total de 42 ECTS. O grau de “Licenciado em Ciência e Engenharia Informática” será conferido aos estudantes que realizem o total de 180 créditos correspondentes aos primeiros 6 semestres curriculares do curso.

9.1. Justification of the total number of credit units and of the duration of the study cycle, based on articles no.8 or 9 (1st cycle), 18 (2nd cycle), 19 (integrated master) and 31 (3rd cycle) of Decreto-Lei no. 74/2006, March 24th:

According to Article 19º of Decreto-Lei no 74/2006, being the case of an integrated study cycle as we propose with 5 year full duration (10 terms), we assign to the study cycle a total of 300 ECTS for obtaining the Master degree. To the master dissertation work is assigned a total of 42 ECTS. A BSc degree will be conferred to the students that acquire the total of 180 ECTS corresponding to the completion of the first six semesters of the programme.

9.2. Metodologia utilizada no cálculo dos créditos ECTS das unidades curriculares:

Para cálculo dos créditos ECTS das UCs foi utilizado como indicador a equiparação de 1 unidade ECTS a 28 horas de trabalho do estudante. O esforço do estudante nas várias componentes de atividade de cada unidade curricular tem sido continuamente aferido e quando necessário reajustado pelos docentes e comissão científica do curso, de forma informada por inquéritos periódicos na FCT UNL, desde o início do processo de Bolonha. Na presente proposta foi de novo analisada a creditação das UC, tendo sido preocupação equilibrar o esforço do estudante entre os vários semestres do curso. Procurou-se também contribuir para o aumento da taxa de sucesso, procedendo a um ajuste da carga do estudante no 1º semestre do curso, e reduzindo o número de unidades curriculares simultâneas (de 5 para 4) nos quatro primeiros semestres do curso. De acordo com

as recomendações gerais da FCT UNL, todas unidades curriculares são medidas em múltiplos de 3 ECTS, por questões de modularidade e flexibilidade.

9.2. Methodology used for the calculation of ECTS credits:

Our calculation of ECTS credits of curricular units was based on the correspondence of 1 ECTS unit to 28 working hours. The student load in the various components of activity of each CU has been continuously monitored and adjusted whenever considered necessary by the teaching staff and program scientific committee, informed by periodic student surveys at FCT UNL, in particular after the installation of the Bologna process. When preparing this proposal, the ECTS for all CUs was again reconsidered, with a special concern to better balance the student load along the several terms of the program. We also sought to contribute for increasing student success rates, by adjusting the student work load in the 1st semester of the course, and reducing the number of concurrent CUs (from 5 to 4) during the four first semesters of the course. According to the general recommendations of FCT UNL, all CU are measured in multiples of 3 ECTS units, for the sake of modularity and flexibility.

9.3. Indicação da forma como os docentes foram consultados sobre o método de cálculo das unidades de crédito:

O esforço do estudante nas várias componentes de atividade de cada unidade curricular do curso de Mestrado Integrado em Engenharia Informática que se propõe, e que em muitos casos resultam de adaptação de UCs que transitaram de atuais cursos de 1º ciclo (Licenciatura em Engenharia Informática) e 2º ciclo (Mestrado em Engenharia Informática), tem sido continuamente aferido e reajustado pelos docentes das mesmas sob a coordenação das Comissões Científicas, de forma informada por inquéritos na FCT UNL, desde o início do processo de Bolonha. Estes inquéritos periódicos auscultam os estudantes sobre o número de horas que consomem nas várias atividades, e informam o processo de ajuste de créditos ECTS. Durante o processo de elaboração das fichas das UC incluídas nesta proposta, os docentes estiveram novamente ativamente envolvidos e auscultados sobre o método de cálculo das unidades ECTS, tendo também sido promovidas reuniões com os estudantes.

9.3. Indication of the way the academic staff was consulted about the method for calculating the credit units:

The student load in the various components of activity of each curricular unit of the Mestrado Integrado em Engenharia Informática now proposed, and that in many cases resulted from updating UCs existing UC in the 1st cycle (Licenciatura em Engenharia Informática) and 2nd cycle (Mestrado em Engenharia Informática), has been continuously monitored and adjusted by the academic staff under the supervision of unit responsible and the program's scientific committees. This process has also been informed by the periodic student surveys carried out at FCT UNL. These surveys consult students about their workload in several academic activities and courses, and are taken into account in the ECTS unit calculation processes. During the process of elaboration of this proposal and of the CU descriptions included, the academic staff and steering committees were again actively involved and consulted about the calculation method, and meetings with students were also organized.

10. Comparação com ciclos de estudos de referência no espaço europeu

10.1. Exemplos de ciclos de estudos existentes em Instituições de referência do Espaço Europeu de Ensino Superior com a duração e estrutura semelhantes à proposta:

O curso proposto compara-se naturalmente com os programas de escolas europeias de referência que visam a formação de engenheiros informáticos de conceção. Por ex, tanto a Universidade Politécnica de Madrid como o Imperial College London oferecem formações de 4 anos. No entanto, a dissertação ou não existe de todo ou apresenta fôlego mais reduzido do que o nosso curso exige, nos termos da Lei. Independentemente da organização formal dos cursos, e sem compromisso da mobilidade preconizada por Bolonha, parece ser consensual a nível europeu que a formação do engenheiro de conceção só se completa em 4-5 anos. Em Portugal, o modelo de mestrado integrado na área de Informática é o adotado na Faculdade de Engenharia da Universidade do Porto, a maioria dos cursos restantes visando a formação de engenheiros de conceção estão organizadas em dois ciclos independentes, numa lógica de continuidade orientada para uma formação coerente em 5 anos, que a presente proposta assume explicitamente.

10.1. Examples of study cycles offered in reference Institutions of the European Higher Education Area with similar duration and structure to the proposed study cycle:

The proposed program naturally compares with programs of reference European schools targeting the education of conception-oriented informatic engineers. E.g., both the Universidade Politécnica de Madrid and the Imperial College London offer 4 year programs. However, the dissertation either does not exist, or targets more limited goals than our program demands, according to the law. Independently of each program's formal

structure, and without compromising the mobility requirement enforced by Bologna, it seems widely accepted in Europe that the education of the conception-oriented engineer is only concluded after 4-5 years. In Portugal, in the field of Computer Science and Engineering, the Faculdade de Engenharia da Universidade do Porto already adopts the integrated 2-cycle master scheme, while the majority of other educational offers is organized in two independent cycles, but in a logic of continuity oriented towards a coherent 5 year education, which our proposal explicitly assumes.

10.2. Comparação com objectivos de aprendizagem de ciclos de estudos análogos existentes em Instituições de referência do Espaço Europeu de Ensino Superior:

Considerando ciclos de estudo (integrados ou 3+2) análogos verifica-se um alinhamento semelhante ao desta proposta, reconhecendo-se uma organização em 3 níveis: básico (1º-3º anos), de consolidação (3º-4º anos) e especialização (4º-5º anos). Em relação aos 2 primeiros níveis existe bastante consenso sobre os objetivos e competências relevantes (recomendações ACM/IEEE). O Imperial College, entre outras escolas, inclui a meio do curso a UC “Industrial Placement”, de objetivos semelhantes à UC “Atividade Prática de Desenvolvimento Curricular”. A natureza do bloco de especialização é mais variada. Existem cursos em que se procura o aprofundamento de uma dada vertente. Outros, como nesta proposta, permitem ao estudante optar entre concentrar a formação numa sub-área ou aprofundar assuntos à sua escolha, numa perspetiva de abrangência. A estrutura dos 1º e 2º ciclos que propomos permite uma natural integração quer com outros ciclos integrados, quer com outros 1º e 2º ciclos europeus.

10.2. Comparison with the intended learning outcomes of similar study cycles offered in reference Institutions of the European Higher Education Area:

Considering similar (integrated or 3+2) programs, we notice the alignment with our proposal, where a three tier structure is recognizable: core (1st-3rd year), consolidation (3rd-4th years), specialization (4th-5th years). There is a wide consensus on the subjects and competencies to cover in the first two tiers (ACM/IEEE). The Imperial College London, among other, includes the “Industrial Placement” unit in the middle of the curriculum, with similar goals as our “Practical Activity for Curricular Development”. The nature of the specialization tier varies a bit: there are programs that enforce a focus in a particular sub-field; others, like our proposal, allow the student to choose between concentrating in a sub-field or to flexibly pick themes, in a perspective of broadening his education. The structure of the 1st and 2nd cycles of our integrated master allows a smooth articulation either with other integrated cycles, or with other autonomous 1st and 2nd cycles offered in Europe.

11. Estágios e Períodos de Formação em Serviço

11.1. e 11.2 Indicação dos locais de estágio

Mapa VII - Protocolos de Cooperação

Mapa VII - N/A

11.1.1. Entidade onde os estudantes completam a sua formação:

N/A

11.1.2. Protocolo (PDF, máx. 100kB):

<sem resposta>

Mapa VIII. Mapas de distribuição de estudantes

11.2. Mapa VIII. Mapas de distribuição de estudantes. Plano de distribuição dos estudantes pelos locais de estágio. (PDF, máx. 100kB)

Documento com o planeamento da distribuição dos estudantes pelos locais de formação em serviço demonstrando a adequação dos recursos disponíveis.

<sem resposta>

11.3. Recursos próprios da Instituição para acompanhamento efectivo dos seus estudantes no período de estágio e/ou formação em serviço.

11.3. Indicação dos recursos próprios da Instituição para o acompanhamento efectivo dos seus estudantes nos estágios e períodos de formação em serviço:

N/A

11.3. Indication of the Institution's own resources to effectively follow its students during the in-service training periods:

N/A

11.4. Orientadores cooperantes**Mapa IX. Normas para a avaliação e selecção dos elementos das Instituições de estágio responsáveis por acompanhar os estudantes****11.4.1 Mapa IX. Normas para a avaliação e selecção dos elementos das Instituições de estágio responsáveis por acompanhar os estudantes (PDF, máx. 100kB)**

Documento com os mecanismos de avaliação e selecção dos monitores de estágio e formação em serviço, negociados entre a Instituição de Ensino e as Instituições de formação em serviço.

<sem resposta>

Mapa X. Orientadores cooperantes de estágio e/ou formação em serviço (obrigatório para ciclo de estudos de formação de professores)**11.4.2. Mapa X. Orientadores cooperantes de estágio e/ou formação em serviço (obrigatório para ciclo de estudos de formação de professores) / External supervisors responsible for following the students activities (mandatory for teacher training study cycles)**

Nome / Instituição ou estabelecimento a que pertence / Institution	Categoria Profissional / Professional Title	Habilitação Profissional / Professional qualifications	Nº de anos de serviço / Nº of working years
--	---	--	---

<sem resposta>

12. Análise SWOT do novo ciclo de estudos**12.1. Apresentação dos pontos fortes:**

Experiência de quase 40 anos do DI FCT UNL no ensino da Engenharia Informática. Elevada empregabilidade de engenheiros informáticos de conceção e banda larga. Todos os docentes a tempo integral doutorados e membros de centros de investigação. Excelentes infraestruturas de ensino e investigação. Integração em redes de investigação e educação avançada, nacionais e internacionais. A qualidade da investigação do Departamento, e o envolvimento dos estudantes nas atividades de investigação (vários graduados de 2º ciclo de Engenharia Informática do DI são admitidos em programas doutorais extremamente seletivos (ex: Carnegie Mellon, Cambridge). Boa ligação com empresas / empregadores, facilitando o possível envolvimento de estudantes em “internships” e dissertações em ambiente académico-empresarial. Excelente apreciação pelos empregadores (nacionais e internacionais) da formação do DI, assim como pelos alunos (inquéritos). Cobertura de competências transversais e “Perfil FCT”.

12.1. Strengths:

Experience of around 40 years of DI FCT UNL in Computer Science and Engineering higher education. High employment rate for broad-band conception-oriented engineers in the Computer Science / Informatics / IT sector. All full time faculty with PhD degree and member of a research unit. Excellent teaching and research infrastructures. Integration in national and international research and educational networks. Quality of the Department research, and involvement of students in research activities (former 2nd cycle graduates admitted in extremely selective PhD programs (ex: Carnegie Mellon, Cambridge). Good connections with firms and industry, which facilitates the possible involvement of students in internships and dissertation work in the context of academy - company / industry collaboration. Excellent appreciation of DI FCT graduates by employers

(national and international), and by the students as well (through surveys). Coverage of transferable skills and the “FCT Profile”.

12.2. Apresentação dos pontos fracos:

Apesar de terem sido sempre recrutados para a formação em Engenharia Informática do DI FCT UNL candidatos com nota de candidatura em geral superior a 13/20, terão que continuar a ser feitos esforços no sentido de atrair cada vez melhores e mais bem informados candidatos para a área de Engenharia Informática. Existe alguma dificuldade, mesmo em algumas grandes escolas a nível internacional, em atrair mais estudantes de topo para a Engenharia Informática, em comparação com outras Engenharias mais clássicas. A situação poderia beneficiar de uma mais correta visibilidade e enquadramento da área no ensino secundário, e na comunicação com a sociedade em geral; apesar dos esforços feitos, existe em muitos casos uma perceção deficiente do que é verdadeiramente a Engenharia Informática pelos estudantes de Ciência e Tecnologia. Existe também um desequilíbrio de género que seria interessante atenuar.

12.2. Weaknesses:

Although we have been able to recruit for our educational offer in Computer Science and Engineering candidates with a application grade generally higher to 13/20, further efforts are needed to promote the attraction of stronger and better informed candidates for the field. There is a certain difficulty, even in the context of strong international universities, to attract more topmost students for Computer Science and Engineering, when compared with certain more classical engineering fields. This situation could be overcome from a better way of giving visibility and awareness of the area in secondary school (particularly in terminal years), and in general public communication: notwithstanding existing efforts, there is in many cases an insufficient understanding of what Computer Science and Engineering really is, as an engineering and science discipline and as a profession. It would also be interesting to attenuate a noticeable gender unbalance.

12.3. Apresentação das oportunidades criadas pela implementação:

Melhor resposta à elevada e sustentada necessidade de profissionais cada vez mais qualificados em Engenharia Informática. Atualmente, apenas a Faculdade de Engenharia da Universidade do Porto oferece formação em Engenharia Informática sob a forma de mestrado integrado. A orientação do programa para a formação integrada de engenheiros informáticos de conceção, facilitando o fluxo de alunos no sentido do 2º ciclo, potencia a coesão e solidez da formação concedida ao longo dos 5 anos, sem prejuízo da mobilidade entre 1º e 2º ciclo prevista na lei. Currículo atualizado com base no ACM/IEEE 2013 e alinhamento com a investigação, desenvolvimento tecnológico, e mercado atual. O envolvimento precoce dos estudantes em atividades de investigação, reforçando a base de recrutamento interna de doutorandos. Melhor exposição dos alunos a atividades de realização prática e/ou interdisciplinar em ambiente externo à escola, permitida pela adoção do “Perfil FCT” referido em 3.1.3 e 3.2.2.

12.3. Opportunities:

Better response to the currently high and sustained demand for increasingly sophisticated professionals in the Computer Science/Informatics field. Presently, only the Faculdade de Engenharia da Universidade do Porto offers an integrated master program in Informatics. The program’s orientation towards the integrated 5 year long education of conception-oriented engineers facilitates the students’ progress towards the 2nd cycle and leverages the cohesion of the program, without compromising mobility between 1st and 2nd cycle, as required by law. Updated curriculum, based on ACM/IEEE 2013, aligned with current research, technological development and market needs. Early involvement of students in research may contribute to increase the candidate basis for our PhD program. Clearer exposition of students to hands-on activities, possibly in interdisciplinary themes and/or externally to the school, supported by the “FCT Profile” mentioned in 3.1.3 and 3.2.2, promotes outreach and integration in society.

12.4. Apresentação dos constrangimentos ao êxito da implementação:

A questão geral de eventual agravamento das dificuldades na gestão flexível e disponibilidade de recursos humanos e materiais, inerentes ao enquadramento institucional e à situação económica global, em particular relativamente ao reforço e renovação de recursos humanos. A diminuição do número de candidatos admitidos com notas de candidatura de nível excelente, que contribuem para catalizar o desempenho dos estudantes em geral e potenciar a subida geral da média de candidatura, que como se refere e explica em 12.2 não acompanha de forma uniforme outras áreas de Engenharia, mesmo a nível global.

12.4. Threats:

The general issue of possible degradation of the ability to flexibly manage human and material resources, intrinsic to our institutional insertion and the global economical situation, in particular with relation to the need of strengthening and renewing human resources. A decrease in the number of admitted candidates with top-notch

grades, who currently contribute to catalyse the performance of the classes in general and to increase the average admission grade, which, as mentioned and explained in 12.2, does not compare favourably with other classical engineering programs, even at a global level.

12.5. CONCLUSÕES:

A presente proposta de curso de Mestrado Integrado em Engenharia Informática consubstancia uma visão atualizada sobre a formação do Engenheiro Informático de conceção (nível E2 de qualificação da Ordem dos Engenheiros).

O objetivo do curso é formar Engenheiros Informáticos de conceção habilitados a desenvolver atividades de projeto, liderança, e inovação, por vezes em contexto de investigação, e com bases para aceder a um 3º Ciclo. O plano curricular está concebido de forma a preparar profissionais para evoluírem com sucesso numa área intensamente sujeita à mudança, como é o caso da Informática. A orientação do programa para a formação integrada de engenheiros informáticos de conceção, facilita o fluxo de estudantes no sentido do 2º ciclo e potencia a solidez e eficácia da formação concedida ao longo de 5 anos. Esta orientação é implementada numa lógica de continuidade, sendo que os eventuais diplomados de 1º ciclo, mesmo não sendo o alvo natural do curso, adquirem uma formação coerente e relevante em termos da mobilidade prevista na lei. Estes objetivos estão alinhados com o projeto educativo, científico e cultural da Escola. É uma aposta estratégica desta continuar a contribuir decisivamente para o desenvolvimento da educação avançada e investigação científica e inovação em Informática, como Escola pioneira na área em Portugal.

O currículo, matérias e competências cobertas encontram-se atualizadas com base nas mais recentes recomendações ACM/IEEE 2013 (draft February 2012), em alinhamento com a investigação, desenvolvimento tecnológico, e as necessidades do mercado atual. Foi realizada uma análise sistemática de cobertura do currículo ACM/IEEE 2013, verificando-se uma cobertura de temas de nível 1 (e mesmo excedendo-a), uma cobertura de mais de 80% de temas de nível 2 (de acordo com a recomendação ACM/IEEE), e tendo sido a cobertura de temas nível 3 determinada com base em análise de relevância das matérias e na preocupação de cobertura em banda larga. O programa potencia o envolvimento dos estudantes em atividades de investigação, reforçando a base de recrutamento interna de doutorandos. O curso proposto oferece aos estudantes oportunidades de atividade no exterior da escola, e também a possibilidade opcional de realizarem a dissertação em ambiente académico-empresarial, no contexto de protocolos de enquadramento e parcerias estabelecidas com empresas. A adequação ao “Perfil FCT”, comum a todos os cursos da escola, favorece o desenvolvimento de várias competências transversais, na área da comunicação, ciência tecnologia e sociedade, e empreendedorismo. A empregabilidade dos cursos de Engenharia Informática da FCT/UNL tem rondado os 99%. Acredita-se que a proposta aqui apresentada oferece uma resposta eficaz, atualizada e oportuna, à sustentada necessidade de profissionais cada vez mais qualificados em Engenharia Informática, sendo compatível com as mais elevadas referências de formação afim na Europa e Estados Unidos.

12.5. CONCLUSIONS:

This program proposal for the Integrated Master in Computer Science and Engineering implements an updated perspective on the education of the conception-oriented Computer Science Engineers / Informatics (level E2 of Ordem dos Engenheiros).

The objective of the program is to educate conception-level Computer Science Engineers prepared to develop project, leadership, and innovation activities, some times in a R&D context, with a basis to pursue a 3th cycle, according to Artº 15 of DL74/2006. The curricular plan was designed to prepare professionals for a successful carrier path in a rapidly changing field. The program’s orientation towards the education of conception-oriented engineers facilitates students’ progress towards the 2nd cycle and strengthens the effectiveness and cohesion of the program along 5 years. These guidelines are implemented in a progressive scheme, so that possible 1st cycle graduates, even if not in the main target of the integrated master, may receive a sufficiently coherent and relevant education useful for mobility purposes, as required by law.

These goals are totally aligned with the educational, scientific and cultural project of FCT/UNL. It is a strong strategical commitment of FCT UNL to continue contributing for the development of advanced education and research in Computer Science and Engineering as a pioneering school in the field in Portugal.

The curriculum, subjects, and competencies conform to the recent recommendations of the ACM/IEEE 2012 (draft of February 2012), well aligned with the modern research, technological development, market and society needs.

We have systematically analyzed the ACM/IEEE 2013 coverage, and observed a total (and even exceeding) coverage of tier 1 topics, and more than 80% coverage of tier 2 topics (thus conforming to ACM/IEEE recs). Our coverage of tier 2 topics was based on our analysis of subject relevance, combined with a concern for breath. The program also promotes the early involvement of students in research, which is expected contribute for increasing the candidate basis for our PhD program. It also offers students several opportunities of involvement in hands-on activities, some in the field, including the optional opportunity of developing their master dissertation work within a collaboration between academia and a company or external institution, in the context of established agreements. The program conforms to the “FCT Curricular Profile”, which favours the emergence of various transferable skills in areas such as communication, science, technology and society, and

entrepreneurship. The employment rate of FCT/UNL's Computer Science and Engineering programs approximates 99%. We strongly believe that this program proposal offers an effective, updated, and timely response to the sustained need for increasingly sophisticated professionals in Computer Science and Engineering, compatible with the highest standards for programs with similar goals in Europe and the USA.