

NCE/19/1901036 — Apresentação do pedido - Novo ciclo de estudos

1. Caracterização geral do ciclo de estudos

1.1. Instituição de Ensino Superior:

Universidade Nova De Lisboa

1.1.a. Outra(s) Instituição(ões) de Ensino Superior (proposta em associação):

1.2. Unidade orgânica (faculdade, escola, instituto, etc.):

Faculdade De Ciências E Tecnologia (UNL)

1.2.a. Outra(s) unidade(s) orgânica(s) (faculdade, escola, instituto, etc.) (proposta em associação):

1.3. Designação do ciclo de estudos:

Engenharia Informática

1.3. Study programme:

Computer Science and Engineering

1.4. Grau:

Licenciado

1.5. Área científica predominante do ciclo de estudos:

Engenharia Informática

1.5. Main scientific area of the study programme:

Computer Science and Engineering

1.6.1 Classificação CNAEF – primeira área fundamental, de acordo com a Portaria n.º 256/2005, de 16 de Março (CNAEF-3 dígitos):

523

1.6.2 Classificação CNAEF – segunda área fundamental, de acordo com a Portaria n.º 256/2005, de 16 de Março (CNAEF-3 dígitos), se aplicável:

481

1.6.3 Classificação CNAEF – terceira área fundamental, de acordo com a Portaria n.º 256/2005, de 16 de Março (CNAEF-3 dígitos), se aplicável:

<sem resposta>

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

180

1.8. Duração do ciclo de estudos (art.º 3 DL n.º 74/2006, de 24 de março, com a redação do DL n.º 65/2018, de 16 de agosto):

3 anos (6 semestres)

1.8. Duration of the study programme (article 3, DL no. 74/2006, March 24th, as written in the DL no. 65/2018, of August 16th):

3 years (6 semesters)

1.9. Número máximo de admissões:

170

1.10. Condições específicas de ingresso.*Provas de Ingresso**Um dos seguintes conjuntos:**(19 Matemática A) OU (04 Economia, 19 Matemática A) OU (07 Física e Química, 19 Matemática A)**Nota mínima de candidatura: 95 pontos**Prova de ingresso: 95 pontos**Fórmula de Cálculo:**Média do Ensino Secundário: 60%**Provas de ingresso: 40%***1.10. Specific entry requirements.***Admission Examinations**One of the following sets:**(19 Matemática A) OU (04 Economia, 19 Matemática A) OU (07 Física e Química, 19 Matemática A)**Minimal application mark: 95 / 200**Admission examination: 95 / 200**Computation Rule:**Secondary School Grade Average: 60%**Admission examinations: 40%***1.11. Regime de funcionamento.***Diurno***1.11.1. Se outro, especifique:**

-

1.11.1. If other, specify:

-

1.12. Local onde o ciclo de estudos será ministrado:*Campus da Faculdade de Ciências e Tecnologia da Universidade NOVA de Lisboa***1.12. Premises where the study programme will be lectured:***Campus da Faculdade de Ciências e Tecnologia da Universidade NOVA de Lisboa***1.13. Regulamento de creditação de formação académica e de experiência profissional, publicado em Diário da República (PDF, máx. 500kB):**[1.13._Reg.459-2020_creditação de competencias_11-05-2020.pdf](#)**1.14. Observações:***A FCT NOVA é pioneira no ensino da Engenharia Informática em Portugal. A presente proposta assenta na experiência do DI FCT NOVA, acumulada em 44 anos, no ensino da Engenharia Informática, complementada com a opinião de consultores externos, e consubstancia uma visão atualizada sobre a formação inicial do Engenheiro Informático de conceção.**O ciclo de estudos proposto, compatível com os princípios de Bolonha, visa conceder aos diplomados uma formação de base em Informática coerente e abrangente, que poderá posteriormente ser aprofundada ao nível do segundo ciclo. O curso permite uma inserção na atividade profissional compatível com o grau de qualificação E1 da Ordem dos Engenheiros, e está alinhada com o estabelecido na escola, para as Engenharias, e com programas afins em várias escolas de referência.**A estrutura do curso baseia-se nas recomendações do ACM/IEEE CS Curriculum 2013 e também nas recomendações da Ordem dos Engenheiros para os cursos de Engenharia.**O curso consiste essencialmente em UC obrigatórias que cobrem (e excedem) as competências definidas para o nível 1*

das recomendações ACM/IEEE acima referidas. No fim do curso é oferecida a UC "Atividade de Desenvolvimento Curricular", de 12 ECTS, que concede experiência de projeto, de iniciação à investigação, ou estágio em empresa, em contexto flexível e variável. 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 Mestrado Integrado em Engenharia Informática (MIEI), do qual o curso proposto deriva - sendo muito semelhante aos três primeiros anos do MIEI - recebeu em 2017 a marca de qualidade EUR-ACE. O EUR-ACE® é um sistema de acreditação, implementado pela ENAEE (European Network for Accreditation of Engineering Education), que fornece um conjunto de padrões que identificam programas na área da Engenharia de alta qualidade na Europa.

1.14. Observations:

FCT NOVA pioneered Computer Science and Engineering (CSE) higher education in Portugal, with the Informatics Department founded 44 years ago. This proposal builds on the accumulated experience of DI FCT NOVA, complemented by the opinion of external consultants, and substantiates an up-to-date view of the initial skills of a Computer Science and Engineering professional with design competences.

The proposed programme, compatible with the Bologna principles, aims to provide graduates with a coherent and comprehensive set of skills, which can then be further developed at the level of the second cycle. The programme allows an insertion in the professional activity compatible with the degree of qualification E1 of the "Ordem dos Engenheiros" (Portuguese Engineering Council) and is aligned with the established practice at FCT NOVA for engineering programmes, with the accreditation procedures of "Ordem dos Engenheiros" in Portugal, and with many similar programmes in reference schools.

The programme structure is based on the recommendations of the ACM / IEEE CS Curriculum 2013 as well as the recommendations of OE for Engineering courses.

The course consists essentially of compulsory CU that cover (and exceed) level 1 of the ACM / IEEE recommendations mentioned above. At the end of the programme, the course "Curriculum Development Activity" (12 ECTS credits) provides, in a flexible and diversified context, project experience, research initiation, or internship in an enterprise. This curricular unit is part of the "FCT Curricular Profile", a differentiating mark of the school, supported by soft skill CU, immersion in practical activities, and multi-disciplinarity.

The Integrated Master in Computer Engineering (MIEI), from which the proposed course derives - being very similar to the first three years of MIEI - received in 2017 the EUR-ACE quality mark. EUR-ACE® is an accreditation system, implemented by the European Network for Accreditation of Engineering Education (ENAEE), which provides a set of standards that identify high quality engineering programmes in Europe.

2. Formalização do Pedido

Mapa I - Aprovação pelo Reitor da NOVA, ouvido o Colégio de Diretores

2.1.1. Órgão ouvido:

Aprovação pelo Reitor da NOVA, ouvido o Colégio de Diretores

2.1.2. Cópia de ata (ou extrato de ata) ou deliberação deste órgão assinada e datada (PDF, máx. 100kB):

[2.1.2._Despachos Reitorais adaptacao assinados pelo Reitor_08-05-2020 7_LEI.pdf](#)

Mapa I - Conselho Científico da FCT NOVA

2.1.1. Órgão ouvido:

Conselho Científico da FCT NOVA

2.1.2. Cópia de ata (ou extrato de ata) ou deliberação deste órgão assinada e datada (PDF, máx. 100kB):

[2.1.2._Dec_CC_LEI.pdf](#)

Mapa I - Conselho Pedagógico da FCT NOVA

2.1.1. Órgão ouvido:

Conselho Pedagógico da FCT NOVA

2.1.2. Cópia de ata (ou extrato de ata) ou deliberação deste órgão assinada e datada (PDF, máx. 100kB):

[2.1.2._Dec_CP_LEI.pdf](#)

Mapa I - Plano de Creditação do Mestrado Integrado em Engenharia Informática

2.1.1. Órgão ouvido:

Plano de Creditação do Mestrado Integrado em Engenharia Informática

2.1.2. Cópia de ata (ou extrato de ata) ou deliberação deste órgão assinada e datada (PDF, máx. 100kB):

[2.1.2._Plano de Creditação_MIEI.pdf](#)

3. Âmbito e objetivos do ciclo de estudos. Adequação ao projeto educativo, científico e cultural da instituição

3.1. Objetivos gerais definidos para o ciclo de estudos:

A Engenharia Informática (EI) é a disciplina técnica, científica, e profissional que permite conceber e construir os sistemas informáticos que desempenham um papel nuclear na sociedade. Os conhecimentos e competências do Engenheiro Informático são essenciais em todas as organizações que desenvolvem serviços, sistemas e produtos. O curso proposto é o 1.º passo na formação de um profissional que irá conceber estes sistemas.

Um licenciado em EI deve possuir um conjunto de competências que lhe permitam:

- Conhecer princípios e técnicas, e desenvolver capacidades analíticas que suportem uma abordagem profissional ao trabalho na área da EI;
- Saber aplicar os conhecimentos na resolução de problemas no âmbito da EI e justificar as opções tomadas.
- Saber pesquisar e integrar conhecimentos, avaliar e comparar soluções criticamente;
- Revelar capacidade para comunicar resultados de forma clara e rigorosa;
- Demonstrar atitudes colaborativas, que suportem a aprendizagem, com autonomia, ao longo da vida.

3.1. The study programme's generic objectives:

Computer Science and Engineering (CSE) is the technical, scientific, and professional discipline that enables the design and implementation of computer systems that play a core role in society. The CSE professional knowledge and skills are essential in all organizations that develop services, systems and products. The proposed course is the first step in training a professional who will design these systems. A CSE graduate must have a set of skills that enable him / her to:

- Know principles and techniques, and develop analytical skills that support a professional approach to work in the field of CSE.
- Know how to apply knowledge in problem solving in the CSE area and justify the choices made.
- Know research and integrate knowledge, evaluate and compare solutions critically.
- Reveal ability to communicate results clearly and accurately.
- Demonstrate collaborative attitudes that support lifelong learning with autonomy.

3.2. Objetivos de aprendizagem (conhecimentos, aptidões e competências) a desenvolver pelos estudantes:

Compreender os fundamentos e o estado-da-arte em Engenharia Informática ao nível de um primeiro ciclo, 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 auto-atualização contínua;

Enfrentar a complexidade de forma proativa, 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 num primeiro nível de qualificação, permitindo a mobilidade de acordo com os princípios de Bolonha.

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

Understand the fundamentals and state of the art in Computer Science and Engineering (CSE) at the level of a first cycle, use mathematical and technical bases, and scientific and engineering methods in the development of professional activities;

To deal with the distinct aspects of computer systems, combining skills geared towards technical development with those of interpretation of the social context in which they operate, in a spirit of continuous self-actualization;

Address complexity proactively by demonstrating creativity.

Have soft skills that allow integration in teams, know how to document goals, solutions and results for experts and non-specialists;

Address ethical and professional aspects, including social and environmental responsibilities;

Report, synthesize, and argue about technical issues, demonstrate critical insight about proposed solutions;

Perform professional activity in CSE at a first level of qualification, allowing mobility according to the Bologna principles.

3.3. Inserção do ciclo de estudos na estratégia institucional de oferta formativa, face à missão institucional e, designadamente, ao projeto educativo, científico e cultural da instituição:

A FCT NOVA 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 Licenciatura em Engenharia Informática constitui o primeiro passo para a formação de Engenheiros com o nível adequado ao desenvolvimento de projeto e de atividades de inovação. Acresce que dado o caráter 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 44 anos, como escola pioneira na área no nosso País. É de referir o alinhamento do curso com o chamado “Perfil Curricular FCT”, que favorece o desenvolvimento de várias competências transversais, na área da comunicação, sociedade, sustentabilidade e transformação digital, 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”. Estas atividades colaborativas estão integradas numa unidade curricular Atividade de Desenvolvimento Curricular (2.º semestre do 3.º ano) em que o estudante pode desenvolver um projeto de dimensão considerável, que além da “Internship” em Empresa já referida, incorpora também uma vertente Projeto de Engenharia e Inovação em que grupos de estudantes desenvolvem um produto que procura resolver um problema relevante para a sociedade de forma inovadora; outra vertente é a de Introdução à Investigação em que o estudante é integrado numa equipa de investigação do centro de investigação NOVA LINCS.*

3.3. Insertion of the study programme in the institutional educational offer strategy, in light of the mission of the institution and its educational, scientific and cultural project:

FCT NOVA 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 areas, including research oriented towards the solution of societal challenges.*
- b) Excellence in teaching, supported by research activities and implemented by academic programmes, competitive at the national and international level;*
- c) A broad basis for interinstitutional participation aimed at integrating 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 to the social development and to the high-quality qualification of human resources.*

The BSc programme now proposed corresponds to the first step in the training of CSE engineers, with a level of competence suitable to project development and innovation activities. Moreover, given the enabling technology characteristics of Computer Science and Engineering (CSE), and its increasing impact in society, we are in the 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 social and economic development. Therefore, FCT NOVA is strategically strongly committed to continue contributing to the development of advanced education and research in CSE, as has been doing for 44 years now, as the pioneering school in the field at the national level. The BSc program is perfectly aligned with the so-called “FCT Curricular Profile”, which enables the emergence of various transferable skills in areas such as communication, society, sustainability and digital transformation, 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 in academic-industrial collaboration.

These collaborative activities are integrated into Curricular Development Practical Activity CU already mentioned, that also incorporates an Engineering and Innovation Project option where groups of students develop a product that seeks to solve a relevant problem for society in an innovative way; another option is Introduction to Research where the student is included in a research team at the NOVA LINCS Research Center.

4. Desenvolvimento curricular

4.1. Ramos, opções, perfis, maior/menor ou outras formas de organização em que o ciclo de estudos se estrutura (a preencher apenas quando aplicável)

4.1. Ramos, opções, perfis, maior/menor ou outras formas de organização em que o ciclo de estudos se estrutura (a preencher apenas quando aplicável) / Branches, options, profiles, major/minor or other forms of organisation (if applicable)

Ramos, opções, perfis, maior/menor ou outras formas de organização em que o ciclo de estudos se estrutura:

Branches, options, profiles, major/minor or other forms of organisation:

<sem resposta>

4.2. Estrutura curricular (a repetir para cada um dos percursos alternativos)

Mapa II -

4.2.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):

4.2.1. Branch, option, profile, major/minor or other (if applicable):

4.2.2. Áreas científicas e créditos necessários à obtenção do grau / Scientific areas and credits necessary for awarding the degree

Área Científica / Scientific Area	Sigla / Acronym	ECTS Obrigatórios / Mandatory ECTS	ECTS Mínimos optativos* / Minimum Optional ECTS*	Observações / Observations
Informática / Informatics	I	123	3	
Matemática / Mathematics	M	30	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	3	0	
Competências Complementares / Transferable Skills	CC	3	0	
Qualquer Área Científica / Any Scientific Area	QAC	0	6	
(7 Items)		171	9	

4.3 Plano de estudos

Mapa III - - 1.º Ano / 1st Year

4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):

4.3.1. Branch, option, profile, major/minor or other (if applicable):

4.3.2. Ano/semestre/trimestre curricular:

1.º Ano / 1st Year

4.3.3 Plano de Estudos / Study plan

Unidade Curricular / Curricular Unit	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS / Observations (5)
Análise Matemática I / Mathematical Analysis I	M	Semestre 1/Semester1	168	T:42; PL:28	6
Álgebra Linear e Geometria Analítica / Linear Algebra and Analytic Geometry	M	Semestre 1/Semester1	168	T:42; PL:28	6
Introdução à Programação / Introductory Programming	I	Semestre 1/Semester1	252	TP:63; PL:21	9
Sistemas Lógicos / Logical Systems	EEC	Semestre 1/Semester1	168	T:28; PL:42	6
Competências Transversais para Ciência e Tecnologia / Soft Skills for Science and Technology	CC	Trimestre 2/Quarter2	80	TP:10; PL:50	3

Análise Matemática II E / Mathematical Analysis II E	M	Semestre 2/Semester2	168	TP:42; PL:28	6
Matemática Discreta / Discrete Mathematics	M	Semestre 2/Semester2	168	T:42; PL:28	6
Arquitetura de Computadores / Computer Architecture	I	Semestre 2/Semester2	252	T:42; PL:28	9
Programação Orientada pelos Objetos / Object-Oriented Programming	I	Semestre 2/Semester2	252	T:42; PL:28	9

(9 Items)

Mapa III - - 2.º Ano / 2nd Year

4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):

4.3.1. Branch, option, profile, major/minor or other (if applicable):

4.3.2. Ano/semestre/trimestre curricular:

2.º Ano / 2nd Year

4.3.3 Plano de Estudos / Study plan

Unidade Curricular / Curricular Unit	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	Semestre 1/Semester1	252	T:42; PL:28	9	
Fundamentos de Sistemas de Operação / Operating Systems Foundations	I	Semestre 1/Semester1	252	T:42; PL:28	9	
Lógica Computacional / Computational Logic	I	Semestre 1/Semester1	168	T:42; TP:28	6	
Física / Physics	F	Semestre 1/Semester1	168	T:42; PL:28	6	
Sociedade, Sustentabilidade e Transformação Digital /	CHS	Trimestre 2/Quarter2	80	TP:42	3	
Bases de Dados / Databases	I	Semestre 2/Semester2	252	T:42; PL:28	9	
Linguagens e Ambientes de Programação / Programming Languages and Environments	I	Semestre 2/Semester2	168	T:42; PL:28	6	
Probabilidades e Estatística D / Probability and Statistics D	M	Semestre 2/Semester2	168	TP:42; PL:14	6	
Teoria da Computação / Theory of Computation	I	Semestre 2/Semester2	168	T:42; PL:28	6	

(9 Items)

Mapa III - - 3.º Ano / 3rd Year

4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):

4.3.1. Branch, option, profile, major/minor or other (if applicable):

4.3.2. Ano/semestre/trimestre curricular:

3.º Ano / 3rd Year

4.3.3 Plano de Estudos / Study plan

Unidade Curricular / Curricular Unit	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
--------------------------------------	---------------------------------------	------------------------	------------------------------------	------------------------------------	------	--------------------------------

Computação Gráfica e Interfaces / Computer Graphics and Interfaces	I	Semestre 1/Semester1	168	T:42; PL:28	6	
Engenharia de Software / Software Engineering	I	Semestre 1/Semester1	252	T:42; PL:28	9	
Inteligência Artificial / Artificial Intelligence	I	Semestre 1/Semester1	168	T:42; PL:28	6	
Redes de Computadores / Computer Networks	I	Semestre 1/Semester1	168	T:42; PL:28	6	
Análise e Desenho de Algoritmos / Design and Analysis of Algorithms	I	Semestre 2/Semester2	168	T:42; PL:28	6	
Atividade de Desenvolvimento Curricular / Curricular Development Activity	I	Semestre 2/Semester2	336	TP:28; PL:28; S:28; OT:10	12	
Sistemas Distribuídos / Distributed Systems	I	Semestre 2/Semester2	168	T:42; PL:28	6	
Programa de Oportunidades / Opportunities Program	I	Trimestre 2/Quarter2	80	OT: 7	3	Optativa / Optional
Unidade Curricular de Bloco Livre A / Unrestricted Elective A	QAC	Semestre 2/Semester2	165	depende da UC escolhida / dependent of choice	6	Optativa / Optional

(9 Items)

Mapa III - - 3.º Ano - Programa de Oportunidades / 3rd Year - Opportunities Program

4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):

4.3.1. Branch, option, profile, major/minor or other (if applicable):

4.3.2. Ano/semestre/trimestre curricular:

3.º Ano - Programa de Oportunidades / 3rd Year - Opportunities Program

4.3.3 Plano de Estudos / Study plan

Unidade Curricular / Curricular Unit	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 Investigação Científica / Undergraduate Research Opportunity Programme	I	Trimestre 2/Quarter2	80	OT:7	3	Optativa / Optional
Programa de Introdução à Prática Profissional / Undergraduate Professional Opportunity Programme	I	Trimestre 2/Quarter2	80	OT: 7	3	Optativa / Optional

(2 Items)

4.4. Unidades Curriculares

Mapa IV - Análise Matemática I

4.4.1.1. Designação da unidade curricular:

Análise Matemática I

4.4.1.1. Title of curricular unit:

Mathematical Analysis I

4.4.1.2. Sigla da área científica em que se insere:

M

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

José Maria Nunes de Almeida Gonçalves Gomes - T:42

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Jorge Manuel Leocádio André - PL:84

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

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

- 1) *Trabalhar com noções elementares de topologia na reta real;*
- 2) *Fazer pequenas demonstrações por indução matemática;*
- 3) *Compreender a noção rigorosa de limite (de sucessões e de funções de variável real) e efetuar o seu cálculo;*
- 4) *Compreender a noção rigorosa de continuidade de funções de variável real e respetivos resultados fundamentais;*
- 5) *Compreender a noção rigorosa de diferenciabilidade, os teoremas de Rolle, Lagrange e Cauchy e aplicações ao cálculo de limites;*
- 6) *Conhecer o Teorema de Taylor e as suas aplicações no estudo de funções;*
- 7) *Conhecer a noção de primitiva e respetivas técnicas de cálculo;*
- 8) *Conhecer a noção de integral de Riemann, respetivas técnicas de cálculo e algumas aplicações;*
- 9) *Ser capaz de analisar a convergência de integrais impróprios.*

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

At the end of this course the student should have acquired knowledge and skills to be able to:

- 1) *Work with elementary notions of topology on the real line;*
- 2) *Make small proofs using mathematical induction;*
- 3) *Understand the definition of limit (for sequences and functions of real variable) and be able to calculate it;*
- 4) *Understand the definition of continuity for functions of real variable and the fundamental associated results;*
- 5) *Understand the definition of differentiability, theorems of Rolle, Lagrange and Cauchy and their applications to the calculation of limits;*
- 6) *Well know the Taylor Theorem and its applications to the analysis of functions;*
- 7) *Understand the notion of indefinite integral and perform the corresponding calculations;*
- 8) *Understand the notion of Riemann integral, the techniques for calculation and some applications;*
- 9) *Be able to study the convergence of improper integrals.*

4.4.5. Conteúdos programáticos:

1. *Topologia elementar da reta real.*
2. *Indução Matemática e sucessões.*
3. *Limites e Continuidade em R.*
4. *Cálculo Diferencial em R.*
5. *Cálculo Integral em R.*

4.4.5. Syllabus:

1. *Basic topology of the real line.*
2. *Mathematical induction and sequences.*
3. *Limits and Continuity in R.*
4. *Differential Calculus in R.*
5. *Integral Calculus in R.*

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Os algoritmos indicativos ao lado dos tópicos referidos remetem para a indexação dos objetivos de aprendizagem acima descritos:

1. *Topologia elementar da reta real. Objetivos de Aprendizagem: 1.*
2. *Indução Matemática e sucessões. Objetivos de Aprendizagem 2,3.*
3. *Limites e Continuidade em \mathbb{R} . Objetivos de Aprendizagem 4.*
4. *Cálculo Diferencial em \mathbb{R} . Objetivos de Aprendizagem: 5,6.*
5. *Cálculo Integral em \mathbb{R} . Objetivos de Aprendizagem: 7, 8, 9.*

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The numbered objectives refer to the above numbered intended outcomes.

1. *Basic topology of the real line. Objective 1.*
2. *Mathematical induction and sequences. Objectives 2, 3.*
3. *Limits and Continuity in \mathbb{R} . Objective 4.*
4. *Differential Calculus in \mathbb{R} . Objectives 5,6.*
5. *Integral Calculus in \mathbb{R} . Objectives 7, 8, 9.*

4.4.7. Metodologias de ensino (avaliação incluída):

O método de ensino consiste no modelo académico Aulas Teóricas/Aulas Práticas. Nas aulas teóricas a matéria é exposta através de resultados justificados, exemplos e aplicações. Exercícios são feitos de forma autónoma pelos alunos e discutidos nas aulas práticas. Alguns exercícios são feitos em sala de aula com orientação docente, sempre visando a autonomização do estudo.

Os dois métodos de avaliação seguidos são:

Avaliação Contínua: realização de três testes de uma hora e trinta minutos igualmente espaçados no semestre e cuja média aritmética fornece a nota final do aluno. Na data de exame, os alunos que o pretendam podem efetuar a melhoria de um dos testes.

Avaliação por Exame: Realização de um exame de três horas composto por três grupos relativos às matérias avaliadas nos três testes da avaliação contínua.

4.4.7. Teaching methodologies (including students' assessment):

The teaching method consists on conference classes and problem solving sessions. On the conference classes, the theory is exposed together with examples and applications. Selected exercises should be done autonomously by the students and are discussed/corrected in the problems solving sessions. Some exercises are discussed directly in the problem solving sessions with guidance by the professor, aiming the autonomization of the students.

We provide two evaluation methods:

Continuous evaluation: Three tests of one and an half hour during the semester whose average provides the final grade.

Evaluation by exam: An exam of three hours composed of three major parts coinciding with the subjects of the three tests.

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

As componentes teóricas necessárias para atingir os objetivos são transmitidas nas aulas teóricas, e consolidadas através da explicitação de exemplos e aplicações. Durante as aulas práticas são realizados ocasionalmente exercícios de natureza mais teórica com vista a um aprofundamento da matéria.

As componentes práticas para atingir os objetivos resultam do trabalho desenvolvido nos turnos práticos, fortemente baseado na interação docente/aluno. As fichas de exercícios são feitas ad-hoc para cada sessão prática e definem o nível de dificuldade de testes e exames.

Existem horários de atendimento ao longo da semana e eventualmente horários extraordinários em período de testes para que os alunos possam beneficiar de um apoio particular dos docentes da UC.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The theoretical basis required to attain the objectives are transmitted during the conference classes. Discussions, examples and counter examples are recurrently used to settle the knowledge. During the problem solving sessions some problems are oriented to a deeper understanding of the theory.

The practical skills are developed during the problem solving sessions, strongly based on the interaction of the students with the teacher. Each problem solving sessions is organized by a sheet of exercises designed for it. They also set the expected level of exercises in tests and exams.

During the week, there is an attending schedule provided by the teachers where students may obtain an individual help.

4.4.9. Bibliografia de consulta/existência obrigatória:

A UC dispõe de um texto de apoio da autoria de José Gonçalves Gomes e de Isabel Azevedo Gomes que consigna as aulas teóricas e é disponibilizado aos alunos.

A text containing the theoretical lectures by José M. Gomes and Isabel A. Gomes is provided to the students.

Outros textos de referência/ other reference text books:

Alves de Sá, A. e Louro, B., Cálculo Diferencial e Integral em \mathbb{R}

Alves de Sá, A. e Louro, B., Cálculo Diferencial e Integral em \mathbb{R} , Exercícios Resolvidos, Vol. 1,2,3
Anton, Bivens and Davis, Calculus ed Wiley.
Campos Ferreira, J., Introdução à Análise Matemática, ed Fundação Calouste Gulbenkian.
Lages de Lima, E., Curso de Análise Vol 1, ed IMPA (projeto Euclides)
Rudin, Principles of Mathematical Analysis, ed Mac Graw Hill

Mapa IV - Álgebra Linear e Geometria Analítica

4.4.1.1. Designação da unidade curricular:

Álgebra Linear e Geometria Analítica

4.4.1.1. Title of curricular unit:

Linear Algebra and Analytic Geometry

4.4.1.2. Sigla da área científica em que se insere:

M

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Herberto de Jesus da Silva - T:42

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Maria Cecília Perdigão Dias da Silva - PL:84

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

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

- Operar com matrizes, caracterizar as matrizes invertíveis e calcular a inversa de uma matriz invertível.*
- Utilizar as matrizes para determinar se um sistema de equações lineares é impossível ou é possível e, neste caso, determinar o conjunto das soluções.*
- Representar uma aplicação linear por uma matriz e determinar, por exemplo, se a aplicação é sobrejetiva, se é injetiva, determinando a característica da matriz.*
- Dada uma matriz quadrada, calcular o seu determinante, os seus valores próprios e respetivos vetores próprios associados.*
- Utilizar as matrizes e determinantes na Geometria Analítica em \mathbb{R}^3 , por exemplo para a determinação de uma equação geral de um plano, a determinação da posição relativa entre 2 rectas (entre 2 planos ou entre 1 reta e 1 plano).*

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

The student is supposed acquire basic knowledge on Linear Algebra. At the end of the curricular unit students should have the following abilities:

- To use matrices in different situations*
- To recognize an invertible matrix*
- To compute the inverse of an invertible matrix*

- To work on systems of linear equations using matrices
- To know the relation between a matrix and a linear function
- To understand the determinant of a square matrix, related results, to compute the eigenvalues and eigenspaces and their applications
- To use matrices, systems of linear equations and the concept of determinant to solve some geometric problems.

4.4.5. Conteúdos programáticos:

- 1 - Matrizes
- 2 - Sistemas de Equações Lineares
- 3 - Determinantes
- 4 - Espaços Vetoriais
- 5 - Aplicações Lineares
- 6 - Valores e Vetores Próprios
- 7 - Produto Interno, Produto Externo e Produto Misto de Vetores em R^3
- 8 - Geometria Analítica em R^3

4.4.5. Syllabus:

- 1 – Matrices
- 2 – Systems of Linear Equations
- 3 – Determinants
- 4 – Vector Spaces
- 5 – Linear Transformations
- 6 – Eigenvalues and Eigenvectors
- 7 - Inner, Vector and Mixed Products in R^3
- 8 – Analytic Geometry in R^3

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

No Capítulo 1 inicia-se o estudo das matrizes e, em particular, caracterizam-se as matrizes invertíveis e deduz-se um método para determinar a inversa de uma matriz invertível. No Capítulo 2 consideram-se os sistemas de equações lineares na forma matricial. No Capítulo 3 apresenta-se a noção de determinante de uma matriz quadrada e algumas propriedades do determinante. Nos Capítulos 4 e 5 são apresentadas e exploradas as noções de espaço vetorial, de aplicação linear e de matriz de uma aplicação linear. No Capítulo 6 estudam-se os valores próprios e vetores próprios de uma matriz (quadrada). Nos restantes capítulos faz-se uma introdução à geometria analítica em R^3 com a utilização das matrizes e determinantes.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

In Chapter 1 we study Matrix Algebra and matrices are used along all the other chapters. In Chapter 2 we work on systems of linear equations using matrices. In Chapter 3 we present the notion of determinant of a square matrix and derive several properties. Along Chapters 4 and 5 we present and study vector spaces, linear functions and matrix representations of a linear function. In Chapter 6 we study eigenvalues, eigenvectors and eigenspaces of a square matrix. In the remaining chapters we present an introduction to Analytic Geometry.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas são lecionados os conceitos e os resultados fundamentais que, na sua maioria, são demonstrados. Ao longo da aula são apresentados exemplos ilustrativos e são propostos exercícios que os alunos deverão resolver autonomamente de forma a consolidar a matéria teórica lecionada. Nas aulas práticas os alunos têm a possibilidade de resolver exercícios e de propor exercícios para resolução de forma a esclarecer as dúvidas surgidas durante o tempo dedicado ao estudo autónomo da matéria. No horário de atendimento docente cada aluno pode, individualmente, esclarecer as suas dúvidas com qualquer um dos docentes da unidade curricular.

*Avaliação: 2 testes (T1 e T2 as classificações obtidas no 1.º e 2.º testes) ou Exame
Nota final: $0,4 \times T1 + 0,6 \times T2 \geq 9,5$ ou Nota do Exame.*

4.4.7. Teaching methodologies (including students' assessment):

Theoretical classes consist on an explanation of the theory which is illustrated by examples. Most results are proven. Practical classes consist on the resolution of some exercises. Some of the exercises are solved in class, the remaining are left to the students as part of their learning process.

*Assessment: 2 tests (T1 and T2 the ratings obtained in the 1st and 2nd tests) or Exam
Final grade: $0.4 \times T1 + 0.6 \times T2 \geq 9.5$ or Exam score.*

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

As componentes teóricas determinadas nos objetivos da unidade curricular são lecionadas nas aulas teóricas onde também se apresentam exercícios práticos para ilustrar conceitos e resultados. A aprendizagem é consolidada com a

componente das aulas práticas, o estudo autónomo do aluno e, se necessário, utilizando o horário de atendimento dos docentes. A frequência na unidade curricular pretende assegurar que os alunos acompanham a matéria. A avaliação de conhecimentos é efetuada através de provas escritas (testes/exames).

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The main concepts specified in the objectives of the course are explained in the theoretical lectures in which some practical problems are also presented as an illustration of concepts or results. Learning is consolidated with the component of the practical classes, the student's self-study and, if necessary, using the office hours of teachers. The frequency in the course aims to ensure that students follow the matter. The assessment is made through written tests (tests / exams).

4.4.9. Bibliografia de consulta/existência obrigatória:

ISABEL CABRAL, CECÍLIA PERDIGÃO, CARLOS SAIAGO, Álgebra Linear, Escolar Editora, 2018 (5.ª Edição).

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. Carvalho, Álgebra Linear e Geometria Analítica, texto de curso ministrado na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Departamento de Matemática da FCT/UNL, 2000.
<http://ferrari.dmat.fct.unl.pt/personal/jvc/alga2000.html>*

E. Giraldes, V. H. Fernandes e M. P. M. Smith, Álgebra Linear e Geometria Analítica, McGraw-Hill de Portugal, 1995.

Mapa IV - Introdução à Programação

4.4.1.1. Designação da unidade curricular:

Introdução à Programação

4.4.1.1. Title of curricular unit:

Introduction to Programming

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

252

4.4.1.5. Horas de contacto:

TP:63; PL:21

4.4.1.6. ECTS:

9

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Maria Armanda Simenta Rodrigues Grueau (Regente) - TP:126; PL:63

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

António Maria Lobo César Alarcão Ravara – TP:126; PL:21
Bernardo Parente Coutinho Fernandes Toninho – TP:126; PL:42
Artur Miguel de Andrade Vieira Dias – TP:126; PL:42
Miguel Jorge Tavares Pessoa Monteiro – TP:63; PL:21

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber: as construções do fragmento coberto de Java; construir uma aplicação no fragmento a partir de uma especificação informal, com a metodologia definida; os componentes e ferramentas básicas de um ambiente de desenvolvimento de software e sua função.

Fazer: desenvolver programas de pequena dimensão, segundo certas convenções; desenvolver algoritmos simples; entender código escrito no fragmento coberto de Java; utilizar ferramentas de programação e interpretar os seus resultados; realizar, em grupo, um mini-projeto de desenvolvimento de software, integrando competências.

Desenvolver: hábitos de trabalho, individuais e em grupo, e de cumprimento de prazos; preocupação com a organização, o rigor e a execução de planos de trabalho.

4.4.4. Intended learning outcomes (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 and their role.

Know-how: 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; Develop as a team, a software development mini-project, using the skills acquired in this course.

Soft-Skills: Develop disciplined work and deadline meeting skills; Develop a concern with rigour and the systematic execution of work plans, following previously defined methods; Develop team work skills.

4.4.5. 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

4.4.5. 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 principle

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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.

4.4.6. Evidence 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.

4.4.7. Metodologias de ensino (avaliação incluída):

O ensino consiste em aulas teórico-práticas e aulas práticas (ambas em laboratório). Nas teórico-práticas a exposição de matéria é intercalada com a resolução de exercícios nos quais os alunos aplicam os conceitos apresentados. Nas práticas, os alunos resolvem exercícios de consolidação dos conceitos apreendidos e praticados durante as aulas teórico-práticas. No mini-projeto final os alunos desenvolvem uma aplicação simples que permite aplicarem toda a matéria aprendida nas aulas. O apoio ao projeto nas 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.

Avaliação:

Teórico-prática: 2 testes (T1 e T2) ou 1 exame (E).

Laboratorial: 2 trabalhos práticos (TP1 e TP2).

Frequência: média (TP1 e TP2) ≥ 9.5

Nota final:

*-Avaliação contínua: $0.2 * T1 + 0.4 * T2 + 0.15 * TP1 + 0.25 * TP2$*

*-Época de Recurso: $0.6 * E + 0.15 * TP1 + 0.25 * TP2$*

Aprovação: nota ≥ 9.5 e ter frequência.

4.4.7. Teaching methodologies (including students' assessment):

There are two lecture and a lab session each week.

In the lectures, all course contents are presented using concrete practical examples. In the laboratory, students design, analyse and implement programs for specific problems by applying the knowledge gained. Students will be evaluated by practical works and two tests.

Evaluation:

Theoretical-practical: 2 tests (T1 and T2) or 1 exam (E).

Laboratory: 2 practical works (TP1 and TP2).

Frequency: medium (TP1 and TP2) ≥ 9.5

Final grade:

*- Continuous evaluation: $0.2 * T1 + 0.4 * T2 + 0.15 * TP1 + 0.25 * TP2$*

*- Time of Resource: $0.6 * E + 0.15 * TP1 + 0.25 * TP2$*

Approval: grade ≥ 9.5 and have frequency.

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

As aulas teórico-práticas são lecionadas em turnos de pequena dimensão (no máximo 28 alunos). Esta topologia de aulas tem várias vantagens. Primeiro, nestas aulas os alunos aplicam imediatamente os conceitos apresentados durante a exposição teórica. Desta forma, qualquer dúvida/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 uma 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 consolidem 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.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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.

4.4.9. Bibliografia de consulta/existência obrigatória:

David J. Eck, Programming Using Java, Online book, <http://math.hws.edu/javanotes>, 2014.

Cay Horstmann, Java Concepts, 7th edition, Wiley, 2014.

Walter Savitch, Java: An Introduction to Computer Science and Programming, 7th edition, Prentice-Hall, 2014.

Especificação das classes do java 8 (API): <https://docs.oracle.com/javase/8/docs/api/>

Mapa IV - Sistemas Lógicos

4.4.1.1. Designação da unidade curricular:

Sistemas Lógicos

4.4.1.1. Title of curricular unit:

Logical Systems

4.4.1.2. Sigla da área científica em que se insere:

EEC

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:28; PL:42

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Luis Filipe dos Santos Gomes - T:28; PL:42

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

4.4.4. Objetivos de aprendizagem (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.

4.4.4. Intended learning outcomes (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.

4.4.5. Conteúdos programáticos:

- *Álgebra de Boole; Tabelas de verdade.*
- *Funções lógicas: Formas canónicas; Simplificação de funções; Mapas de Karnaugh; Método de Quine-McCluskey.*
- *Sistemas de numeração.*
- *Aritmética binária: Soma e subtração; Compl. para 2 e para 1; Multiplicação e divisão.*
- *Circuitos combinatórios elementares.*
- *Elementos de memória biestáveis: flip-flop JK, D T.*
- *Circuitos sequenciais: Noção de sistema síncrono e assíncrono; Registos; Desenho expedito de contadores.*
- *Máquinas de estado síncronas: Diagramas de estado, Síntese.*
- *Memórias; Dispositivos de lógica programável.*
- *Introdução a arquiteturas de transferência entre registos: decomposição em partes de controlo e de dados; introdução aos microprocessadores.*

4.4.5. Syllabus:

- *Boolean algebra; 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.*
- *Memory elements: Flip-flop JK, D and T.*
- *Sequential circuits: Synchronous and asynchronous circuits' concepts; Registers. Counter design.*
- *Synchronous state machines: State diagrams; Synthesis.*
- *Memories; Programmable logic devices.*
- *Introduction to register transfer architectures: decomposition in control and data parts; introduction to microprocessors.*

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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 que serão posteriormente utilizadas pelos alunos.

Os conteúdos programáticos previstos encontram-se estruturados em quatro grupos, nomeadamente conceitos introdutórios, análise de módulos combinatórios, análise de circuitos sequenciais e análise de circuitos digitais de reduzida/média complexidade obtidos através da sua decomposição em parte de controlo e parte de dados.

Os objetivos de aprendizagem identificados permitem ir verificando a obtenção de competências ao longo dos conteúdos apresentados.

4.4.6. Evidence 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 that will later be regularly used by students.

The provided syllabus are structured into four groups, including introductory concepts, combinatorial analysis modules, circuit analysis sequential, and small / medium complexity digital circuit analysis obtained by its decomposition into control and data parts.

The identified learning outcomes allow checking achievement of specific skills, as it is possible a direct association between learning outcomes and those groups within the proposed syllabus.

4.4.7. Metodologias de ensino (avaliação incluída):

Considera-se uma aula teórica de 2h/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.

Avaliação: 2 testes individuais (30%+30%) ou exame final (60%) (componente teórica), conjunto de perguntas com apoio do moodle (10%) (avaliação contínua) e a realização de um mini-projeto em grupos de 2 ou 3 alunos (30%)(componente prática).

4.4.7. Teaching methodologies (including students' assessment):

Contact with students is accomplished in one lecture of 2h / 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.

Evaluation is guaranteed through 2 individual tests (30%+30%) or final exam (60%) (theoretical component), a set of short questions with moodle support (10%) (continuous assessment) and conducting one mini-project in groups of 2 or 3 students (30%)(practical component).

4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos 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.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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.

4.4.9. Bibliografia de consulta/existência obrigatória:

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

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

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

***** Em português:*

Circuitos Digitais e Microprocessadores - Herbert Taub - McGraw-Hill - ISBN 0-07-066595-8

Mapa IV - Competências Transversais para as Ciências e Tecnologias

4.4.1.1. Designação da unidade curricular:

Competências Transversais para as Ciências e Tecnologias

4.4.1.1. Title of curricular unit:

Soft Skills for Science and Technology

4.4.1.2. Sigla da área científica em que se insere:

CC

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

80

4.4.1.5. Horas de contacto:

TP:10; PL:50

4.4.1.6. ECTS:

3

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Ruy Araújo da Costa - TP:10; PL:50

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

<sem resposta>

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):*No final desta UC um aluno deve ser capaz de:*

- escrever o seu 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;
- gerir adequadamente o tempo e trabalhar em equipa;
- compreender a importância da liderança;
- utilizar folhas de cálculo Excel produzindo gráficos com facilidade;
- utilizar no Excel o Solver e ser capaz de programar funções e macros 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;
- perceber a importância do domínio básico do Inglês na área de Ciências e Tecnologia (CT);
- comunicar adequadamente na área de CT.

4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):*After this course, any student should be able to:*

- write his (her) CV and prepare for a job interview;
- understand the importance of taking steps to make his (her) CV more appealing;
- understand how important Psychometric Testing is when accessing the job market;
- manage time adequately and be able to carry out team work effectively;
- understand the importance of leadership;
- use Excel spreadsheets and be able to represent data in graphs;
- use Excel's Solver and be able to program functions and macros 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;
- understand the importance of English in the Science and Technology area;
- communicate adequately in the Science and Technology area.

4.4.5. Conteúdos programáticos:

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

4.4.5. Syllabus:

- 1 - Curriculum Vitae, Job interview and Psychometric testing.
- 2 - Time management, team work and leadership.
- 3 - Advanced use of Excel spreadsheets.

- 4 - *Bibliographic research and critical analysis of scientific information.*
- 5 - *Communicating in Science and Technology.*

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

A 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 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, o 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.*

4.4.6. Evidence 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.

Each of the five topics in this unit 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.*

4.4.7. Metodologias de ensino (avaliação incluída):

- *Em cada semana há 3 aulas práticas que totalizam 10h (2+4+4h);*
 - *Em cada semana há uma aula 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 UC. 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.*
- A avaliação de conhecimentos é feita em 3 áreas: Participação (com um peso de 20%), Trabalho Individual (40%) e Trabalho de Grupo (40%).*

4.4.7. Teaching methodologies (including students' assessment):

- *In each week there are three practical session with a total of 10 hrs (2+4+4 hrs);*
 - *In each week there is a 2h theoretical-practical session that 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.*
- Assessment involves: Participation (20%), Individual Work (40%) and G Group Work (40%).*

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

- 1 - *Discute-se a forma e o conteúdo de um CV. Os alunos analisam entrevistas simuladas. Comenta-se os vários aspetos relevantes (p.ex., CV, vestuário, apresentação, dicção). Reflete-se 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.*
- 2 - *Aborda-se a Gestão do Tempo no contexto universitário e no contexto profissional. Analisa-se as vantagens e desvantagens do Trabalho em Equipa. Analisa-se as características relevantes de um Líder e a sua importância.*
- 3 - *Utiliza-se o Excel no contexto da representação gráfica de funções. Apresenta-se a Formatação Condicional. Introduce-se a utilização de Tabelas Dinâmicas. Apresenta-se os Comandos de Contagem e de Estatística Básica no Excel. Aborda-se a Procura Vertical de Informação ("PROCV"). É feita uma aplicação do Solver com a Otimização de uma função. É feita uma introdução ao módulo de Visual Basic do Excel, que inclui a definição de funções e macros em VB.*
- 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 - *Os alunos são sensibilizados para a importância do domínio básico da Língua Inglesa. Os alunos obtêm formação sobre a comunicação escrita e oral na área de C&T.*

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

- 1 - *CV writing and presentation is discussed. Students analyse simulated job interviews and reflect on the relevant aspects of a job interview. Students are made aware of the importance to make their CV more appealing throughout their university years. Students go through a batch of psychometric tests, using moodle e-learning platform.*
- 2 - *Time Management is addressed in a university context as well as in a job context. Advantages and disadvantages of*

group work are analyzed. Leader's characteristics are addressed, as well as the importance of leadership.

3 - Students are requested to draw graphs of functions using Excel. Conditional Formatting is presented. Students use Pivot Tables and learn Counting commands and Basic Statistics commands. Students learn how to "look for" information (Vlookup). Solver is introduced to optimize a function. Visual Basic in Excel is presented and students learn how to define functions and macros.

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 re presented.

5 – The importance of using English in the Science and Technology (ST) area is stressed out. Students acquire skills in written and oral presentations in the ST area.

4.4.9. Bibliografia de consulta/existência obrigatória:

- Costa, R., Kullberg, J., Fonseca, J., Martins; N., “Competências Transversais para Ciências e Tecnologia – FCT/UNL” (2013).

Mapa IV - Análise Matemática II E

4.4.1.1. Designação da unidade curricular:

Análise Matemática II E

4.4.1.1. Title of curricular unit:

Mathematical Analysis II E

4.4.1.2. Sigla da área científica em que se insere:

M

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

TP:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Ana Luísa da Graça Batista Custódio – TP:42

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Jorge Manuel Leocádio André – PL:56

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

No final desta unidade curricular o estudante terá adquirido conhecimentos, aptidões e competências que lhe permitem (i) reconhecer algumas famílias de equações diferenciais e de problemas de valor inicial de primeira ordem e dominar técnicas analíticas e numéricas associadas à respetiva resolução; (ii) compreender o conceito de limite e continuidade em funções de várias variáveis, sendo capaz de fazer a sua análise analítica e aplicando-os em métodos numéricos para a resolução de equações; (iii) compreender os conceitos de diferenciabilidade e de derivada direcional em funções de várias variáveis, conseguindo fazer a sua análise e utilizando-os na definição de algoritmos de otimização numérica; (iv) dominar os conceitos de integração dupla e tripla, sendo capaz de efetuar técnicas de cálculo analítico e numérico.

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

At the end of this course the student will have acquired knowledge and skills that will enable him (i) to recognize some families of first order differential equations and initial value problems and master analytical and numerical techniques used in the corresponding solution; (ii) understand the concept of limit and continuity in functions of several variables, being able to analyse it resorting to an analytical approach and apply it in numerical methods to the solution of some equations; (iii) understand the concepts of differentiability and directional derivative of functions of several variables, performing the corresponding analysis and using them in the definition of algorithms for numeric optimization; (iv) master the concepts of double and triple integration, being able to perform analytic and numeric calculus techniques.

4.4.5. Conteúdos programáticos:

1. Equações Diferenciais Ordinárias (EDOs) de primeira ordem
2. Limites e continuidade de funções de várias variáveis
3. Cálculo diferencial em funções de várias variáveis
4. Extremos
5. Integrais Duplos
6. Integrais Triplos

4.4.5. Syllabus:

1. First order Ordinary Differential Equations (ODEs)
2. Limits and continuity of functions of several variables
3. Differential calculus of functions of several variables
4. Extremes
5. Double integration
6. Triple integration

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Os tópicos abordados constituem uma programa básico de Análise Matemática para funções de várias variáveis, com uma pequena introdução às equações diferenciais de primeira ordem e a alguns algoritmos de Análise e Otimização Numérica.

Os pontos 1-2 do programa cumprem os objetivos (i)-(ii), respetivamente. O objetivo (iii) é atingido com os pontos 3-4 do programa. Finalmente, os conteúdos programáticos 5-6 permitem satisfazer o objetivo (iv).

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The covered topics are a basic program in Mathematical Analysis for functions of several variables, with a brief introduction to first order differential equations and to some algorithms of Numerical Analysis and Optimization.

Bullets 1-2 of the syllabus allow to achieve the outcomes (i)-(ii), respectively. Outcome (iii) is reached with bullets 3-4 of the syllabus. Finally, bullets 5-6 provide outcome (iv).

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas são participadas, com exposição oral dos conceitos e metodologias, devidamente complementada com exemplos. Nas aulas práticas serão feitas resoluções de problemas e implementação em computador dos métodos estudados. Eventuais dúvidas poderão ser esclarecidas no decurso das aulas ou em sessões individuais marcadas com os professores.

Os alunos necessitam de assistir a um mínimo de 2/3 das aulas práticas lecionadas para se submeterem a avaliação. A avaliação contínua é baseada em dois testes e um trabalho computacional. Se um aluno não obtiver aprovação através de avaliação contínua poderá vir a obtê-la num exame de recurso.

4.4.7. Teaching methodologies (including students' assessment):

Theoretical classes are participative, with oral presentation of concepts and methodologies, complemented with examples. Practical classes correspond to problem solving sessions and to the implementation of the numerical methods studied. Specific student difficulties will be addressed during classes or in individual sessions scheduled with the professor. Students need to attend a minimum of two thirds of the classes in order to be evaluated. Continuous evaluation is based on two tests and one computational project. If a student does not obtain approval through continuous evaluation, he can try it in an additional exam.

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

A apresentação dos conceitos e métodos numéricos, apoiada com exemplos ilustrativos, pretende motivar os estudantes para a relevância dos tópicos estudados e desenvolver a sua capacidade de aplicar, de forma adequada, os métodos estudados. A componente de implementação computacional permite a resolução de problemas mais desafiantes, com a consequente interpretação crítica dos resultados obtidos.

Os alunos terão oportunidade de testar estas capacidades nas aulas, com o apoio de um professor, ou em estudo individual, comparecendo eventualmente a sessões de atendimento individual, em caso de dificuldades.

A avaliação da unidade curricular incide sobre a aquisição das competências referidas.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

In classes, concepts and numerical methods are presented and illustrated with examples, motivating students for the relevance of the corresponding study, and developing student's ability to properly apply the studied methods. The component related with the numerical implementation allows to address more challenging problems, with the consequent critic interpretation of the results obtained.

Students can test these skills in classes, under the supervision of a professor, or by themselves, scheduling individual appointments with professors, in case of difficulties.

The evaluation of the curricular unit focuses in the acquisition of the mentioned capabilities.

4.4.9. Bibliografia de consulta/existência obrigatória:

1. H. Anton, I. Bivens, S. Davis, *Cálculo, volume II*, Astmed editora, 2005.
2. T. Apostol, *Calculus, volume II*, John Wiley & Sons, 1969.
3. R. Burden, D. Faires, *Numerical Analysis, 9th edition*, Brooks-Cole Publishing, 2011.
4. I. Griva, S. G. Nash, A. Sofer, *Linear and Nonlinear Optimization, second edition*, SIAM, 2009.
5. J. Stewart, *Calculus*, Brooks/Cole Publishing Company, 2005.

Mapa IV - Matemática Discreta**4.4.1.1. Designação da unidade curricular:**

Matemática Discreta

4.4.1.1. Title of curricular unit:

Discrete Mathematics

4.4.1.2. Sigla da área científica em que se insere:

M

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

João Jorge Ribeiro Soares Gonçalves de Araújo - T:42; PL:28

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

<sem resposta>

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Familiarização com tópicos de Matemática Discreta que são relevantes em vários contextos, nomeadamente para uma formação em Ciências da Computação.

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

The goal of this course is to introduce the students to topics and methods of discrete mathematics relevant in computer science.

4.4.5. Conteúdos programáticos:

Parte 1 – Conjuntos, relações e funções

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 – Grafos e Aplicações

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

4.4.5. Syllabus:

Part 1 - Sets, relations and functions

- 1. Sets: representations and basic operations; power set; cardinality*
- 2. Binary relations*
- 3. Functions: bijections; composition and inverse*

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 - Graphs and applications

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

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

O conteúdo programático da unidade curricular é concebido de forma a que, por um lado, os tópicos apresentados sigam a sua ordem natural e que, por outro lado, conduzam aos objetivos da unidade curricular. A seleção dos tópicos foi coordenada com o departamento de informática.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The syllabus is conceived in a way that, on the one hand, the items of the given programme are in their systematic order, and on the other hand, that they lead in a natural way to the objectives of the lecture course.

4.4.7. Metodologias de ensino (avaliação incluída):

A metodologia de ensino inclui aulas teóricas e práticas participadas e trabalho do aluno. Modo de avaliação de conhecimentos: exame final ou avaliação teórica-prática (dois testes). Para aceder aos testes é necessário (para cada um deles) fazer PRÉ-INSCRIÇÃO.

4.4.7. Teaching methodologies (including students' assessment):

The teaching methodology, comprising class-room presentations and work of the student, follows the successful tradition of mathematic teaching in the spirit of Felix Klein. There will be three test during the term and a final exam. The students are required to subscribe to each test/exam, at the CLIP.

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

A combinação de apresentações do professor com trabalho desenvolvido pelo aluno é clássica no ensino matemático e/ou teórico na área da Computação.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The combination of presentations of the professor with own work by the student is the principal guarantee for the success of mathematical and/or theoretical teaching in the area of Computation.

4.4.9. Bibliografia de consulta/existência obrigatória:

- [1] R. Johnsonbaugh, *Discrete Mathematics*, Prentice Hall Inter., 1997
- [2] T. S. Blyth e E. F. Robertson, *Sets and Mappings*, Chapman and Hall, 1986
- [3] N. L. Biggs, *Discrete Mathematics*, Oxford Science Publ., 1994
- [4] K. A. Ross e C. R. B. Wright, *Discrete Mathematics*, Prentice Hall Inter., 1999
- [5] R. J. Wilson e J. J. Watkins, *Graphs an Introductory Approach*, Wiley, 1990
- [6] S. Lipschutz, *Set Theory and Related Topics*, Mc Graw-Hill, 1964
- [7] D.M. Cardoso, J. Szymanski e M. Rostami, *Matemática Discreta*, Escolar Editora, 2009
- [8] A. J. Franco de Oliveira, *Teoria de Conjuntos*, Escolar Editora, 1989
- [9] C. André e F. Ferreira, *Matemática Finita*, Universidade Aberta, 2000

Mapa IV - Arquitetura de Computadores

4.4.1.1. Designação da unidade curricular:

Arquitetura de Computadores

4.4.1.1. Title of curricular unit:

Computer Architecture

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

252

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

9

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Vítor Manuel Alves Duarte - T:84

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Hervé Miguel Cordeiro Paulino - PL:84
Paulo Orlando Reis Afonso Lopes - PL:84
Maria Cecília Farias Lorga Gomes - PL:84

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Conhecimentos

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.

Aptidões e competências

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

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

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.*

Competencies and aptitudes

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

4.4.5. 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.*

4.4.5. 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.*

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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 de exercícios práticos assim como exercícios de programação nas aulas práticas. A aprendizagem e a avaliação inclui ainda uma forte componente de programação em projetos e trabalhos de casa. Todos 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 e realçar a forma como podem beneficiar o desempenho do software.

4.4.6. Evidence of the syllabus 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, solving practical exercises, and with the knowledge obtained through programming exercises in lab classes. Learning and evaluation includes also 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.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas explicar-se-ão e discutir-se-ão os sucessivos tópicos do programa da UC. 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 diversos trabalhos práticos de programação, cujo objetivo é aplicar e consolidar os conceitos que foram aprendidos nas aulas teóricas. Após concluírem os trabalhos práticos de programação, os alunos

devem também discutir com os docentes o comportamento dos programas desenvolvidos e o seu encadeamento com os conceitos aprendidos ao longo da UC.

Avaliação:

*prática (NP) e frequência (obter $\geq 9,5$ a pelo menos 3 dos 5 trabalhos propostos);
teórica (NT): dois testes ou exame (escritos, sem consulta);*

em caso de frequência e NT $\geq 8,5$ a nota final é 20% NP + 80% NT

4.4.7. Teaching methodologies (including students' assessment):

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 beginning of the lecture.

During lab classes students will develop a series of practical programming assignments, whose objective is to apply and consolidate the concepts that were learned in lecture. After concluding each lab 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:

*practical (NP) and frequency assessment (the student needs ≥ 9.5 in 3 of 5 homeworks);
theoretical (NT): two midterms or final exam (written, closed books);*

if admitted to evaluation and NT ≥ 8.5 the final grade is 20% NP + 80% NT

4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos 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 UC.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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.

4.4.9. Bibliografia de consulta/existência obrigatória:

Computer Systems: A Programmer's Perspective, 2/E, Randal E. Bryant, David R. O'Hallaron, Prentice Hall, 2011

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

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

Mapa IV - Programação Orientada pelos Objetos

4.4.1.1. Designação da unidade curricular:

Programação Orientada pelos Objetos

4.4.1.1. Title of curricular unit:

Object Oriented Programming

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

252

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

9

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Carla Maria Gonçalves Ferreira - T:84; PL:28

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Miguel Carlos Pacheco Afonso Goulão - PL:112

Rui Pedro da Silva Nóbrega - PL:112

4.4.4. Objetivos de aprendizagem (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 perspetiva 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.

4.4.4. Intended learning outcomes (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.

4.4.5. 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*

4.4.5. 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*

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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.

4.4.6. Evidence 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.

4.4.7. 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. Nos laboratórios, 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.

As componentes da avaliação incluem a componente teórica; e a componente da nota relativa aos trabalhos práticos e participação do aluno nas aulas. Os trabalhos práticos são realizados em grupos de 2 alunos durante o semestre, e participação dos alunos nas aulas. A nota teórica é constituída por um exame final.

Nota final:

O aluno com frequência obtém aprovação se a nota final (NF) e a nota teórica (NT) ≥ 9.5 .

*$NF = 0.6*NT + 0,05*NA + 0,10*NTP1 + 0,25*NTP2$*

em que:

- NA - participação nas aulas;
- NTP1 - nota trabalho prático 1;
- NTP2 - nota trabalho prático 2.

4.4.7. Teaching methodologies (including students' assessment):

Lectures consist of 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 applying the concepts discussed in the previous lectures.

The evaluation include a theoretical component and a component of the note related to practical work and student participation in classes. Practical assignments are carried out in groups of 2 students during the semester. The theoretical grade consists of a final exam.

Final grade:

The student obtains approval if the final grade (NF) and the theoretical grade (NT) are greater than or equal to 9.5. The final grade is calculated using the formula:

$$NF = 0.6*NT + 0.05*NA + 0.10*NTP1 + 0.25*NTP2$$

on what:

- NA is the grade for class participation;
- NTP1 is the grade of the practical work 1;
- NTP2 is the grade of the practical work 2.

4.4.8. 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 as soft skills enunciados na definição dos objetivos de aprendizagem.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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 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.

4.4.9. Bibliografia de consulta/existência obrigatória:

1. Cay Horstmann, *Java Concepts: for Java 7 and 8, Sixth Edition*, Wiley, 2010.
2. F. Mário Martins, *Java 8 - POO + Construções Funcionais*, FCA, 2017.
3. *Slides de Programação Orientada pelos Objectos (disponibilizados gradualmente na página moodle)*
4. David J. Eck, *Introduction to Programming Using Java, Seventh edition*, 2016.

Mapa IV - Algoritmos e Estruturas de Dados

4.4.1.1. Designação da unidade curricular:

Algoritmos e Estruturas de Dados

4.4.1.1. Title of curricular unit:*Algorithms and Data Structures***4.4.1.2. Sigla da área científica em que se insere:**

I

4.4.1.3. Duração:*Semestral / Semester***4.4.1.4. Horas de trabalho:**

252

4.4.1.5. Horas de contacto:*T:42; PL:28***4.4.1.6. ECTS:**

9

4.4.1.7. Observações:*<sem resposta>***4.4.1.7. Observations:***<no answer>***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Luís Manuel Marques da Costa Caires (sem horas de contacto)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Fernanda Maria Barquinha Tavares Vieira Barbosa - T:84; PL:28**João Miguel da Costa Magalhães - PL:84**Carla Maria Gonçalves Ferreira - PL:112***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Saber**1.Complexidade assintótica.**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.**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.**16.Comunicação oral.***4.4.4. Intended learning outcomes (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.**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.

4.4.5. Conteúdos programáticos:**A. Introdução à Análise de Algoritmos****B. Introdução à Recursividade**

*Divisão e conquista.
Função-memória.*

C. Tipos Abstratos de Dados

*Pilha.
Fila.
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.*

4.4.5. Syllabus:**A. Algorithm Analysis****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.

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Os objetivos 1 e 6 são endereçados no cap. A do Programa. Os objetivos 1, 6, 11 e 13 são ainda endereçados ao longo de todo o semestre, na exposição dos alunos às estruturas de dados e no desenvolvimento de exemplos e do projeto, durante a prática.

Os objetivos 2 e 7 são endereçados nos cap. C e D.

Os objetivos 2,3, 8, 9, 11 e 13 são endereçados no cap. D do Programa. Cada estrutura de dados é apresentada com exemplos de implementação, de execução e discutindo as categorias de problemas relevantes.

Os objetivos 4 e 10 são endereçados nos cap. B (onde se discutem às várias técnicas de desenho de algoritmos) e D (onde se estuda a aplicação destas técnicas em estruturas de dados recursivas relevantes) do Programa. Estes objetivos são também considerados no estudo dos algoritmos de ordenação recursivos (Cap. E).

O objetivo 5 é endereçado no cap. E do Programa.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

Objectives 1 and 6 are addressed in chap. 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 while developing examples and the course project.

Objectives 2 and 7 are addressed in chap. C and D.

Objectives 2, 3, 8, 9, 11 and 13 are addressed in chap. D of the syllabus, as each data structure is presented with implementation and execution examples, and by discussing the relevant categories of problems.

Objectives 4 and 10 are addressed in chap. B (where the considered algorithm design techniques are discussed) and D (where the application of the techniques to recursive data structures is considered) of the syllabus. These objectives are also relevant in the study of recursive sorting algorithms (chap. E).

Objective 5 is addressed in chap. E of the syllabus.

4.4.7. 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.

4.4.7. Teaching methodologies (including students' assessment):

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.

4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos 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).

4.4.8. Evidence of the coherence between the teaching methodologies and the 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).

4.4.9. Bibliografia de consulta/existência obrigatória:

Referências Principais

*Michael T. Goodrich and Roberto Tamassia. Data Structures and Algorithms in Java (6th edition). John Wiley & Sons, 2015.
Mark Allen Weiss. Data Structures and Algorithm Analysis in Java (4th edition). Addison-Wesley, 2013.*

Referência Mais Avançada

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

4.4.1.1. Designação da unidade curricular:

Fundamentos de Sistemas de Operação

4.4.1.1. Title of curricular unit:

Operating Systems Foundations

4.4.1.2. Sigla da área científica em que se insere:

/

4.4.1.3. Duração:*Semestral / Semester***4.4.1.4. Horas de trabalho:**

252

4.4.1.5. Horas de contacto:*T:42; PL:28***4.4.1.6. ECTS:**

9

4.4.1.7. Observações:*<sem resposta>***4.4.1.7. Observations:***<no answer>***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Pedro Abílio Duarte de Medeiros - T:42; PL:28***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Paulo Orlando Reis Afonso Lopes - T:42**Hervé Miguel Cordeiro Paulino - PL:112**Maria Cecília Farias Lorga Gomes - PL:112***4.4.4. Objetivos de aprendizagem (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.

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

Know: -services provided by operating systems (OS), their user and programming interfaces; - OS abstractions for program execution and resource management; - principles of concurrent programming and process coordination; - principles of design and internal organization of an operating system.

Apply: - Relate the theoretical and practical aspects and improve the ability to perform practical laboratory work; - Use the environment and tools to support the development and management of program execution at the operating system interface level and with operating system call programming models involving concurrency, communication and synchronization between processes.

4.4.5. Conteúdos programáticos:

1. Introdução aos sistemas de operação (SO): Serviços do sistema: ficheiro e processo como principais abstrações suportadas, chamadas ao sistema.

2. Gestão dos CPUs: suporte da abstração de processo pelo SO; processos leves: Algoritmos de escalonamento dos CPUs

3. Gestão da memória: espaço de endereçamento de um processo; transformação de endereços reais em virtuais usando páginas; memória virtual usando paginação; algoritmos de substituição de páginas.

4. Programação concorrente: fundamentos, comunicação e sincronização em memória partilhada e em memória distribuída. Processos leves - API pthreads.

5. Ficheiros e dispositivos de entrada/saída: organização e operações do sistema de ficheiros; características dos

dispositivos que suportam o sistema de ficheiros; consistência do sistema de ficheiros. Gestão de dispositivos de entrada / saída. Estrutura de um device driver.

6. Máquinas virtuais

4.4.5. Syllabus:

1. *Introduction to operating systems (OS): System services: file and process as the main abstractions, system calls.*
2. *CPU management: support of process abstraction by the OS; lightweight processes: CPU scheduling algorithms.*
3. *Memory management: process address space; transformation of virtual addresses into real addresses using pages; virtual memory using pagination; page replacement algorithms.*
4. *Concurrent programming: fundamentals, communication and synchronization in shared memory and distributed memory. Lightweight Processes - pthreads API.*
5. *Input / output files and devices: file system organization and operations; characteristics of devices that support file systems; file system consistency. Management of input / output devices. Structure of a device driver.*
6. *Virtual Machines*

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

A unidade curricular tem por objetivo permitir a aprendizagem dos aspetos essenciais dos sistemas de operação nas duas vertentes: prestação de serviços às aplicações e gestão dos recursos. O primeiro aspeto é atingido através das sessões laboratoriais em que o estudante desenvolve e/ou adapta programas em C, e usa ferramentas de observação da utilização de recursos pelo sistema. O segundo objetivo é atingido usando simuladores existentes ou desenvolvendo pequenos programas que correm em modo-utilizador mas ainda assim permitem experimentar diferentes políticas de gestão de recursos.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

Learning the essential roles of operating systems: application services and resource management. The first aspect is achieved through laboratory sessions in which the student develops and / or adapts C programs, and uses system resource observation tools. The second objective is achieved by using existing simulators or developing small programs that run in user mode but still allow them to experiment with different resource management policies.

4.4.7. 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 por objetivo permitir ao estudante i) desenvolver aplicações que usam chamadas de sistema num ambiente controlado (máquinas virtuais), onde por vezes são requeridos privilégios de administração e ii) usar ferramentas de monitorização que requerem esses mesmos privilégios.

Avaliação: Componente Teórica (CompT) + Componente Laboratorial (CompL).

Frequência, garantida por "CompL": Nota \geq a 8,5 valores.

Avaliação Teórica: 2 testes, 1 exame de recurso

Nota CompT = 0,5 T1 + 0,5 T2

Para obter aprovação por testes: CompT \geq 8,5

Avaliação Laboratorial: 3 trabalhos individuais; 1 Projecto (P)

Nota da Componente Laboratorial (CompL) = 0,4(NTP1+NTP2+NTP3)/3 + 0,6*NP*

Para obter aprovação: CompL \geq 8,5

Nota final NF = 0,7(CompT ou nota do exame) + 0,3*CompL*

NF \geq 9,5

4.4.7. Teaching methodologies (including students' assessment):

The lectures present the main topics and discuss the most relevant questions. Students should, preferably before class, study the recommended bibliography.

Lab sessions are intended to enable the student to i) develop applications that use system calls in a controlled environment (virtual machines) where administration privileges are sometimes required and ii) use monitoring tools that require such elevated privileges.

Evaluation: Theoretical-Practical Component (CompT) + Laboratory Component (CompL).

"Frequência", granted by "CompL" grading: Grade \geq 8,5 (max. 20).

CompT grading: 2 mid-term tests, 1 final exam

Grade of CompT = 0,5 T1 + 0,5 T2

To succeed with mid-term tests: CompT \geq 8,5

Laboratory grading: 3 individual small-sized work assignments; 1 group Project (P) for 2 students
Grade of the Laboratory Component : $CompL = 0,4(NTP1+NTP2+NTP3)/3 + 0,6*NP$*
To succeed: $CompL \geq 8,5$
Final grade $NF = 0,7(CompT \text{ or final exam grade}) + 0.3*CompL$*
 $NF \geq 9,5$

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

O curso tem por objetivo permitir a aprendizagem pelo estudante dos fundamentos de sistemas de operação. Este objetivo é atingido através das sessões teóricas e das laboratoriais. Estas, ao explorarem três vertentes, programação, simulação, e uso de ferramentas do SO permitem consolidar, aprofundar, e dotar o estudante de conhecimentos práticos úteis para serem posteriormente desenvolvidos quer em unidades curriculares mais avançadas, quer em cursos de curta duração de cariz empresarial, como os oferecidos pela indústria.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The course enables the student to learn the fundamentals of operating systems. This objective is achieved through theoretical and laboratory sessions. By exploring the three aspects, programming, simulation, and use of OS tools, students consolidate, deepen, and acquire practical knowledge that will be further enhanced both in more advanced curricular units and short-term entrepreneurial courses, such as those offered by the industry.

4.4.9. Bibliografia de consulta/existência obrigatória:

MAIN reference:

Remzi Arpac-Dusseau, Andrea Arpac-Dusseau, Operating Systems: Three Easy Pieces, 2015 v0.9,
<http://pages.cs.wisc.edu/~remzi/OSTEP/>*

COMPLEMENTING references:

Modern Operating Systems, 4th Ed, A. Tanenbaum, H. Bos, Pearson, 2014

Computer Systems: A Programmer's Approach, R. Bryant, D. O'Hallaron, Pearson, 3rd Ed 2015

The C Programming Language, B. W. Kernighan, D. M. Ritchie, 2nd Edition, Prentice Hall, 1988

Mapa IV - Física

4.4.1.1. Designação da unidade curricular:

Física

4.4.1.1. Title of curricular unit:

Physics

4.4.1.2. Sigla da área científica em que se insere:

F

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; TP:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

António Carlos Simões Paiva - T:42 TP:28

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

<sem resposta>

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

No final da unidade é esperado que os estudantes consigam:

- Relacionar os conhecimentos aprendidos com o meio que os rodeia.*
- Identificar as características físicas de um problema.*
- Formular o conjunto de equações necessárias à resolução de um problema com base na identificação do ponto anterior.*
- Perante um problema ter capacidade crítica para avaliar o resultado obtido bem como as suas unidades.*
- Ter adquirido capacidade e autonomia na interpretação e resolução de um problema.*

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

At the end of the lecture course, students are expected to:

- relate the fundamental and applied concepts in physics to daily life problems involving classical mechanics.*
- identify the physical formulation of a given problem.*
- write down the set of equations needed to obtain a final value, according to the formulation above.*
- face a problem with capability of assessing the final result and units.*
- have gained capability to deal on their own with the interpretation and solving of a problem.*

4.4.5. Conteúdos programáticos:

Capítulo 0 Forças
Capítulo 1 Oscilações
Capítulo 2 Ondas
Capítulo 3 Carga Elétrica
Capítulo 4 Campos Elétricos
Capítulo 5 Lei de Gauss
Capítulo 6 Potencial elétrico
Capítulo 7 Capacitância
Capítulo 8 Corrente e resistência
Capítulo 9 Circuitos
Capítulo 10 Campos Magnéticos
Capítulo 11 Campos magnéticos devido a correntes
Capítulo 12 Indução e Indutância
Capítulo 13 Equações de Maxwell; Magnetismo da matéria
Capítulo 14 Ondas eletromagnéticas
Capítulo 15 Interferência
Capítulo 16 Difração

4.4.5. Syllabus:

Chapter 0 Forces
Chapter 1 Oscillations
Chapter 2 Waves
Chapter 3 Electric Charge
Chapter 4 Electric Fields
Chapter 5 Gauss' Law
Chapter 6 Electric Potential
Chapter 7 Capacitance
Chapter 8 Current and Resistance
Chapter 9 Circuits
Chapter 10 Magnetic Fields
Chapter 11 Magnetic Fields Due to Currents
Chapter 12 Induction and Inductance
Chapter 13 Maxwell's Equations; Magnetism of Matter
Chapter 14 Electromagnetic Waves
Chapter 15 Interference
Chapter 16 Diffraction

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

São apresentados os conceitos fundamentais necessários para descrever sistemas físicos em vibração e a propagação de ondas. São também introduzidas as definições e convenções necessárias ao estudo dos sistemas elétricos. Seguidamente, os conceitos introduzidos são utilizados no estudo de circuitos elétricos constituídos por fontes de alimentação, resistências e condensadores.

Na segunda parte do semestre é discutido o fenómeno do magnetismo, e são apresentados os conceitos fundamentais necessários para o estudo de campos magnéticos estáticos e variáveis no tempo. No final da unidade curricular são abordados os fenómenos de interferência e difração de ondas eletromagnéticas que servem também como uma apresentação de fenómenos quânticos.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

In the first part of the semester is presented the necessary fundamental concepts to describe physical vibrating systems wave propagation. They are also introduced the definitions and conventions necessary for the study of electrical systems. After, the concepts introduced are used in the study of electrical circuits composed by power supplies, resistors and capacitors.

In the second part of the semester is discussed the magnetism phenomenon, and is presented the fundamental concepts necessary for the study of static time varying magnetic fields. At the end of the course interference and wave diffraction are discussed and quantum effects follow up.

4.4.7. Metodologias de ensino (avaliação incluída):

O ensino desta disciplina decorre em aulas teóricas e aulas teórico-práticas. Nas aulas teóricas são discutidos os vários temas da matéria com demonstração experimental de vários temas. Nas aulas teórico-práticas são resolvidos problemas de aplicação de toda a matéria e é feita periodicamente avaliação prática.

Avaliação:

Componente Teórica (CT): 2 testes

Componente Prática (CP): resolução de um problema pelo grupo de trabalho. Esta resolução traduz-se numa classificação (CC). CP será a média aritmética arredondada às unidades das classificações obtidas nas quatro CC mais altas.

Classificação final (CF) dos estudantes com avaliação prática é dada por: $CF = CT \times 0.7 + CP \times 0.3$.

4.4.7. Teaching methodologies (including students' assessment):

The teaching of this subject takes place in theoretical and theoretical-practical classes. In theoretical classes the different subjects are discussed. In the theoretical-practical classes the students solve problems about all subjects and are regularly evaluated.

Evaluation:

Theoretical Component (CT): 2 tests

Practical Component (CP): resolution of a problem by the working group. This resolution translates into a classification (CC). CP will be the arithmetic mean rounded to the units of the classifications obtained in the four highest CC.

Final classification (CF) of students with practical evaluation is given by: $CF = CT \times 0.7 + CP \times 0.3$.

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

As componentes teóricas necessárias para atingir os objetivos de aprendizagem são ministradas nas aulas teóricas, com o apoio adicional dos docentes nas aulas práticas e horários de atendimento de alunos, caso se justifique. A aquisição destes conhecimentos é avaliada nas provas escritas (testes/exames). As componentes práticas necessárias para atingir os objetivos de aprendizagem são desenvolvidas em todas as formas de horas de contacto: nas aulas teóricas através da análise e discussão de problemas-tipo; nas aulas práticas através da discussão de problemas. A avaliação destas competências é assegurada por avaliações práticas realizadas nessas aulas.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The theoretical deliverables are provided in the lectures with extra support from demonstrations .. Students are evaluated on these performances through written tests/exams. Students skills are acquired in lectures and practical classes. In the former the contents are analysed, whereas in the latter they are discussed with problem's solving. The evaluation process in both components is achieved through written examination and through evaluation process in the practical classes.

4.4.9. Bibliografia de consulta/existência obrigatória:

Halliday, D., Resnick, R., & Walker, J., Fundamentals of Physics, Wiley. (Any edition will do, latest is 11th.)

or

M. Alonso and E. Finn, Physics, Addison-Wesley, 1992

Mapa IV - Lógica Computacional

4.4.1.1. Designação da unidade curricular:

*Lógica Computacional***4.4.1.1. Title of curricular unit:***Computational Logic***4.4.1.2. Sigla da área científica em que se insere:**

I

4.4.1.3. Duração:*Semestral / Semester***4.4.1.4. Horas de trabalho:**

168

4.4.1.5. Horas de contacto:*T:42; PL:28***4.4.1.6. ECTS:**

6

4.4.1.7. Observações:*<sem resposta>***4.4.1.7. Observations:***<no answer>***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Pedro Manuel Corrêa Calvente Barahona- T:84***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Joaquim Francisco Ferreira da Silva - PL:84**Jorge Carlos Ferreira Rodrigues da Cruz - PL:84**Nuno Miguel Cavalheiro Marques - PL:56***4.4.4. Objetivos de aprendizagem (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*

4.4.4. Intended learning outcomes (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*

4.4.5. Conteúdos programáticos:

1. Lógica Proposicional

1.1. Sintaxe:

- *Definição indutiva de linguagem proposicional*

1.2. Semântica:

- *Tabelas de verdade e álgebra de Boole*
- *Valoração e estrutura de interpretação: satisfação*
- *Validade e consequência lógica; equivalência*

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 1.^a ordem*
- *Tradução de descrições em língua natural*
- *Variáveis livres e substituição*

2.2. Semântica:

- *Valoração e estrutura de interpretação: satisfação*
- *Algoritmos de Decisão e Sistemas de Prova:*

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

3. Indução Matemática

4.4.5. Syllabus:

1. Propositional Logic

1.1. Syntax:

- *Inductive definition of propositional language*

1.2. Semantics:

- *Truth tables and Boolean algebra*
- *Valoration and interpretation structure: satisfaction*

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*

2.3. Deductive systems and Decision Algorithms

- *Natural deduction: Introduction and elimination rules*
- *Resolution: Clausal form, Skolemisation, Unification*

3. Mathematical Induction

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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.

4.4.6. Evidence 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 an 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, overiewing the representational and reasoning limitations of these logics (d).

4.4.7. Metodologias de ensino (avaliação incluída):

O programa é lecionado em aulas teóricas e práticas. Nas primeiras são apresentados os conceitos básicos e as técnicas de demonstração relevantes, ilustrados com vários exemplos.

Nas aulas práticas são resolvidos problemas que obrigam a resolver problemas simples, mas que os obriguem a explorar as técnicas ensinadas.

A avaliação de conhecimentos, inclui 4 testes individuais teóricos, sendo a nota final obtida pela média das 4 componentes. A nota dos testes teóricos pode ser substituída pela nota no exame final.

4.4.7. Teaching methodologies (including students' assessment):

The program is taught in theoretical and laboratorial classes. In the first the basic concepts and relevant demonstration techniques are presented, illustrated with several examples.

In the practical classes, the proposed problems are simple, but require the use and explorations of the mateial taught in the theoretical classes.

Students assessment is composed of 4 individual theoretical tests, the final grade being obtained by the average of the 4 components. The grade of the theoretical tests can be replaced by the grade in the final exam.

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

Os objetivos da unidade curricular identificam o conhecimento de linguagens proposicionais e de primeira ordem (sintaxe e semântica) bem como a capacidade de determinar a validade de raciocínios lógicas, através de demonstrações formais. A metodologia seguida expõe os alunos a vários exercícios desta natureza e sua posterior avaliação, o que exige o treino de raciocínio abstrato e rigoroso, igualmente esperado nesta Unidade Curricular.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The objectives of this curricular unit identify the acquisition of knowledge of propositional and first order languages (syntax and semantics) as well as the ability to determine the validity of logical reasoning, through formal proofs.

The methodology presented exposes the students to several exercises of this nature and their subsequent evaluation, which requires the training of abstract and rigorous reasoning, also expected in this Course Unit.

4.4.9. Bibliografia de consulta/existência obrigatória:

Main textbook:

- *Language, Proof, and Logic, David Barker-Plummer, Jon Barwise, John Etchemendy, Center for the Study of Language and Information; 2nd edition, October 2011.*

Additional reading:

- *Mathematical Logic: a course with exercices. 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.*

Mapa IV - Sociedade, Sustentabilidade e Transformação Digital

4.4.1.1. Designação da unidade curricular:

Sociedade, Sustentabilidade e Transformação Digital

4.4.1.1. Title of curricular unit:

Society, Sustainability and Digital Transformation

4.4.1.2. Sigla da área científica em que se insere:

CHS

4.4.1.3. Duração:

Trimestral / Trimester

4.4.1.4. Horas de trabalho:

80

4.4.1.5. Horas de contacto:

TP:42

4.4.1.6. ECTS:

3

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

José Luís Câmara Leme - TP:42

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

<sem resposta>

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Objetivos: levar os alunos a questionarem-se sobre as relações entre ciência, tecnologia, em particular tecnologia digital, ambiente e sociedade e suas implicações para um futuro sustentável e crescentemente informatizado.

Aquisição de conhecimentos: compreender a estrutura da tecnociência e sua relação com os contextos económico, político, social e cultural; compreender as inter-relações entre ciência, tecnologia e sociedade; compreender a natureza sistémica, holística e transdisciplinar das questões de sustentabilidade; compreender os princípios e resultados do processo de transformação digital.

Aquisição de competências: perspetivar o relacionamento entre ciência, tecnologia e sociedade e suas interações com o ambiente e sustentabilidade; desenvolver o sentido de ética e responsabilidade social e ambiental; relacionar a prática profissional com uma cidadania crítica e consciente; compreender o processo de transformação digital e as suas implicações.

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

Objectives: to lead students to ask themselves about the relationship between science, technology, in particular digital technology, environment and society, and its implications for a sustainable and increasingly computerized future.

Specific capabilities:

(i) knowledge acquisition: understanding the structure of technoscience and its relationship with the economic, political, social and cultural contexts; master the interrelationships between science, technology and society; understand the principles and results of the digital transformation process.

(ii) acquisition of skills: to envision the relationship between science, technology and society and their interactions with the environment and sustainability; develop the sense of ethics and social and environmental responsibility; relate professional practice to the practice of critical and conscious citizenship; understand the digital transformation process and its social and individual implications.

4.4.5. Conteúdos programáticos:

Módulo Sociedade:

- 1. Globalização e Desafios Climáticos*
- 2. Mobilidade e Justiça*
- 3. Cibersegurança*
- 4. Melhoramento Humano/ Human Enhancement*

Módulo Sustentabilidade:

Visões de futuro e caminhos de sustentabilidade - limites do crescimento e implicações dos padrões de produção e consumo; crescimento verde e decrescimento sustentável. Pensamento sistémico para os Objetivos de Desenvolvimento Sustentável (ODS) partindo das forças motrizes e analisando as implicações ambientais, sociais e económicas.

Módulo Transformação Digital:

Abordar a forma como as tecnologias digitais transformam o mundo atual e investigar sobre o futuro digital, incluindo aspetos sociais. Serão considerados exemplos no trabalho, aprendizagem, lazer e organização social.

4.4.5. Syllabus:*Society Module*

1. *Globalization and Climate Challenges*
2. *Mobility and Justice*
3. *Cybersecurity*
4. *Human Enhancement*

Sustainability:

Sustainability visions and pathways Limits to growth limits and Spaceship Earth; implications of production and consumption patterns, green growth and sustainable degrowth proposals.

Systems Thinking for the SDGs, starting from the driving forces and analyzing its environmental, social and economic implications.

Digital Transformation Module

Address how digital technologies transform the current world and research the digital future, including social aspects. Examples of work, learning, leisure and social organization will be considered.

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Tendo em conta que os objetivos da disciplina são levar os alunos a interrogarem-se sobre a natureza e a extensão das relações entre ciência, tecnologia e sociedade no mundo atual, estimulando a sua reflexão crítica no contexto da sua futura experiência profissional e de cidadania, escolheu-se um conjunto de tópicos considerados críticos para esta reflexão. Estes tópicos são abordados a partir da contemporaneidade, mas densificados com uma perspetiva histórica que dê aos alunos uma visão diacrónica e dinâmica das relações entre ciência tecnologia e sociedade. Os tópicos foram escolhidos tendo em conta a sua pertinência atual e a vontade de cobrir um leque de áreas diversificado, mas passível de serem estabelecidas pontes e diálogos entre os vários temas. As experiências individuais dos alunos são valorizadas e o debate é encorajado.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

Given that this course aims at unveiling the nature and extent of the relationship between science, technology and society, thus stimulating students to engage in a critical reflection about their future professional practice and citizenship, we chose a set of topics we deem critical to this discussion. These topics are approached from a contemporary perspective but include a historical perspective that allows students a diachronic and dynamic perspective of the relations between science, technology and society. The topics are chosen taking into account their relevance, the need for covering a diversified range of areas, and the possibility to establish bridges and dialogues between the various themes. The individual experience of the students is valued and the debate is encouraged.

4.4.7. Metodologias de ensino (avaliação incluída):

A UC está organizada em três Módulos. Para toda a UC os grupos são constituídos por 5 ou 6 alunos.

Sociedade

Este módulo é constituído por 4 temas. Os alunos fazem apenas um tema. A pesquisa realizada pelo grupo será apresentada sob a forma de um Pecha Kucha. Horas de contacto: 21h.

Sustentabilidade

Exercício de modelação participada sobre os ODS em que os alunos desenvolvem um diagrama causal recolhendo informação em estudo autónomo para substanciar o modelo e discutir medidas. Avaliação: apresentação dos trabalhos utilizando o diagrama causal como suporte da narrativa. Horas de contacto: 12h.

Transformação Digital

A avaliação deste módulo será feita através da apresentação de um poster por grupo, em sessão pública. Cada poster deve incluir um exercício de sistematização de uma tecnologia digital, da transformação que provoca e do impacto futuro, de acordo com os temas indicados. Horas de contacto: 12h.

4.4.7. Teaching methodologies (including students' assessment):

The curricular unit is organized into three Modules. For the whole curricular unit the groups consist of 5 or 6 students.

Society

This module consists of four themes. Students assist only one themes. The research conducted by the group is presented

in the form of a Pecha Kucha. Contact hours: 21h

Sustainability

Building a vision of a sustainable future. Participatory modeling exercise on SDGs in which students develop a causal loop diagram and collect information in autonomous study to substantiate the model and discuss measures. Evaluation: presentation of the works using the causal loop diagram to support the narrative. Contact hours: 12h

Digital transformation

The evaluation of this module will be done through the presentation of one poster per group, in a public session. Each poster must include an exercise in systematizing digital technology, the transformation it causes and the future impact, according to the suggested themes. Contact hours: 12h.

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

As metodologias de ensino visam sensibilizar os alunos para os tópicos da disciplina através de uma estratégia de envolvimento dos alunos na compreensão ativa dos vários temas, usando elementos que lhes sejam familiares, nomeadamente filmes, documentários e peças de literatura. Uma vez estabilizados estes elementos, que permitem aos alunos o manuseamento de um conjunto de conceitos básicos, introduzem-se elementos novos que, assim, são acomodados no quadro já sedimentado. Finalmente, toda a estratégia de ensino visa estimular a análise crítica das relações contemporâneas entre ciência, tecnologia, sociedade, transformação digital e sustentabilidade no sentido de estimular a responsabilidade social e ética dos futuros cientistas e engenheiros.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The teaching method aims at involving students in the topics of the course promoting an active understanding of the various topics, by using familiar knowledge to them in particular movies, documentaries and books. Once these elements are stabilized, thus allowing students to handle a set of basic concepts, we introduce new elements that should be accommodated in the framework already settled. Finally, the whole teaching strategy aims to stimulate critical analysis of the relationship between science, technology, society, digital transformation, and sustainability and the development of a social and ethical consciousness among these scientists and engineers to be.

4.4.9. Bibliografia de consulta/existência obrigatória:

*Peter Singer, One world - the ethics of globalization; New Haven & London: Yale University Press, 2002.
Manjikian, Mary. Cybersecurity Ethics: an Introduction, Routledge, 2016
Julian Savulescu e Nick Bostrom, Human Enhancement, Oxford University Press, 2009
Mimi Sheller, Mobility Justice. The Politics of Movement in an Age of Extremes. London; Brooklyn, NY: Verso, 2018
Meadows, D. H., Thinking in systems: A Primer. Earthscan. 2008.
Robert, Costanza, and Kubiszewski Ida, eds. Creating a sustainable and desirable future: Insights from 45 global thought leaders. World Scientific, 2014.
Aligning the Organization for Its Digital Future, Gerald C. Kane, Doug Palmer, Anh Nguyen Phillips, David Kiron, and Natasha Buckley, MIT Sloan Management Review, July 26, 2016
Achieving Digital Maturity, Gerald C. Kane, Doug Palmer, Anh Nguyen Phillips, David Kiron, Natasha Buckley, October 01, 2017, MIT Sloan Management Review
Artigos/Research Papers (ACM DL and other sources).*

Mapa IV - Bases de Dados

4.4.1.1. Designação da unidade curricular:

Bases de Dados

4.4.1.1. Title of curricular unit:

Databases

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

252

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

9

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):*Carlos Augusto Isaac Piló Viegas Damásio (Regente) - T:84***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Ricardo João Rodrigues Gonçalves - PL:112**Nuno Miguel Cavalheiro Marques - PL:56**Joerg Matthias Knorr - PL:56***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Pretende-se dotar os alunos das bases necessárias à conceção, construção e análise de bases de dados relacionais.***Saber**

- *O modelo Entidade-Relação e o modelo relacional de dados, e os fundamentos do modelo objeto/relacional*
- *Álgebra relacional*
- *Os fundamentos da linguagem SQL*
- *Normalização de bases de dados com base em dependências funcionais e multivalor*
- *Noções de modelação de dados em XML, e de mecanismos para manipulação desses dados*

Fazer

- *Modelar numa base de dados relacional, um problema de dimensão média*
- *Criar uma base de dados em SQL para problema médio implementando em SQL todos os mecanismos necessários à integridade dos dados*
- *Implementar uma interface simples para manipulação da base de dados*
- *Formular perguntas complexas em SQL*

Soft Skills

- *Trabalhar em equipa*
- *Modelar problemas*
- *Elaborar relatório de análise, desenho e implementação de uma solução*
- *Gestão do tempo*

4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):*To provide the basis for the modeling, implementation, analysis and manipulation of relational databases.***Knowledge**

- *To know the entity-relationship model and the relational data model, as well as the foundations of the object-relational model*
- *Relational algebra*
- *Foundations of the SQL language*
- *Normalization of data bases using functional and multi-valued dependencies*
- *Basics of data modeling in XML and manipulation mechanisms for semi-structured data*

Know-how

- *To model a medium sized real problem in a relational database*
- *To create a database in SQL for the above problem, implementing mechanisms for guaranteeing referential integrity*
- *To implement a simple interface for manipulation of the database*
- *To write complex SQL queries*

Soft skills:

- *Team work in collaboration*
- *Problem modeling capabilities*
- *Write an analysis report, design and implement a solution*
- *Manage time and deadlines*

4.4.5. Conteúdos programáticos:*Introdução aos sistemas de bases de dados**Modelos de dados*

- Modelo ER
- Modelo Relacional

Normalização de Bases de Dados

- Dependências funcionais e multi-valor
- Formas normais: 3.^a, 4.^a e de Boyce-Cood

Linguagens de manipulação de bases de dados

- Álgebra relacional
- Linguagem SQL
- Outras linguagens

Integridade de Bases de Dados

- Integridade de referência
- Asserções e triggers
- Segurança e autorizações

Interação com Bases de Dados

- Embedded SQL, ODBC, JDBC
- Segurança e autorizações
- Transações

Discussão de outros modelos de bases de dados

- Bases de dados objetos/relacional
- Bases de dados dedutivas
- XML

4.4.5. Syllabus:*Introduction to Database Management Systems**Data models*

- Entity-Relationship model
- Relational model

Relational databases normalization

- Functional and multi-valued dependencies
- Normal forms: 3rd, 4th and Boyce-Cood

Database manipulation languages

- Relational Algebra
- SQL query and manipulation language
- Other languages

Database integrity

- Referential integrity
- Assertions and triggers

Interaction with databases

- Embedded SQL, ODBC, JDBC
- Security and authorisations
- Transactions

Discussion about other database models

- Object/relational databases (and the SQL case).
- Deductive databses
- XML

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Dotam-se os alunos de conhecimentos e prática na modelação e interrogação de bases de dados relacionais, com experiência na utilização de um sistema real. São confrontados com modelos alternativos não-relacionais para comparar as des/vantagens de cada um deles.

Tratam-se os aspetos de desenho e modelação recorrendo a diagramas ER e sua tradução para bases de dados

relacionais. Leciona-se complementarmente a teoria para aprofundar todos os conceitos, nomeadamente as diversas formas de redundância com as quais se podem deparar e soluções/compromissos para as eliminar. A importância dos aspetos de integridade dos dados são salientados e reforçados ao longo da UC. A UC introduz os alunos à álgebra relacional utilizada no ensino a um nível avançado da linguagem SQL.

Abordam-se tópicos relevantes da utilização de uma base de dados em ambiente real (transações e segurança). É desenvolvido em equipa um projeto de uma base de dados em Oracle e interface em APEX.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The students acquire knowledge and practice in modelling and querying relational databases, experimenting with a widely adopted real system. Alternative non-relational models are used to compare the dis/advantages of each of them.

The unit demonstrates design and modelling with ER diagrams and their translation to relational databases. In compliment, students learn the theory to strengthen all concepts, in particular the different forms of redundancy as well as the solutions/compromises to eliminate them. The importance of data integrity is fostered and reinforced in the classes. Relational algebra is introduced and used to support teaching of the SQL query language at an advanced level.

Relevant practical topics of the daily use of a database system are discussed (e.g. security and transactions). A team project is developed by students using Oracle including the user interface in APEX.

4.4.7. Metodologias de ensino (avaliação incluída):

A UC tem aulas teóricas apoiadas em slides com exemplos. As práticas têm fichas e guiões para realizar durante as aulas e em estudo autónomo. O planeamento das teóricas incluem sempre espaço para análise e discussão de problemas de testes anteriores. Os estudantes utilizam o sistema Oracle 18c (última versão) e APEX tendo disponível um servidor devidamente configurado para utilização nos laboratórios e por VPN a partir do exterior.

A avaliação é constituída por dois testes (ou exame) e um projeto prático em equipa, com pesos respetivamente de 60% e 40% da nota final. É necessário ter 10 valores na componente teórica para obter aprovação. O projeto tem duas fases em que primeiro é proposto pelo grupo o tema e modelo de base de dados a implementar e depois a entrega final com relatório escrito e código. O projeto é realizado essencialmente fora das aulas, justificando os 9ECTS desta UC. A nota do projeto é individual e resulta da discussão com o grupo.

4.4.7. Teaching methodologies (including students' assessment):

The lectures are supported in slides with substantial examples. The lab sessions have guides and work assignments to be done in classes and as homework. The lectures' scheduling include always time to analyse and discuss previous midterm problems. The students use the latest version of Oracle (18c) and APEX, and they have available a server for use in the laboratories as well as a VPN connection to use it outside FCT facilities.

Assessment is composed by two tests (or exam) and a team project work, worthing 60% and 40% of the final grade, respectively. It is necessary to obtain 10 out of 20 in the theoretical component to be approved. The project has two phases where first is proposed the theme and model of the database to be implemented, and secondly the final delivery with a written report and code. The project is mostly developed outside classes, justifying the 9 ECTS of this unit. The project grade is individual and results from discussion with the team.

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

Os métodos de ensino desta UC têm algumas particularidades que permitem aos alunos adquirir os conhecimentos para realização do trabalho prático o mais cedo possível. Nomeadamente, as aulas teóricas semanalmente têm 2 blocos de 2 horas permitindo no final de 5 semanas ter toda a matéria teórica para realizar o trabalho prático. Desta forma, as aulas teóricas também terminam mais cedo permitindo aos alunos focarem-se no seu projeto no final do semestre.

O primeiro teste ocorre antes da primeira entrega do projeto e onde se avalia toda a teoria do desenho e modelação de bases de dados. Os alunos têm os conhecimentos assimilados e estudados para poderem propor um modelo adequado para a sua base de dados. Pelo contrário, o segundo teste é realizado após a entrega do projeto podendo-se avaliar os conhecimentos da linguagem SQL após os estudantes a terem realmente utilizado. Os saberes e fazeres estão assim perfeitamente entrosados.

Os alunos utilizam o sistema Moodle com prazos estritos e com penalizações significativas por entregas fora de horas. Os projetos são discutidos oralmente e demonstrados pelos elementos do grupo, complementado pela avaliação dos relatórios e teste ao código submetido. A interface também é testada e avaliada. Atingem-se assim os objetivos de soft skills desta UC, em especial gestão do tempo e promoção do trabalho de equipa responsável.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The teaching methodologies of this curricular unit have some characteristics that allow students to acquire the necessary knowledge to start their project early in the semester. Namely, the lectures have two weekly blocks of two hours each, allowing that after 5 weeks students have been exposed to all the theory to execute the project. In this way, lectures also

finish before the end of the semester such that students can focus on the project.

The first midterm occurs before the first delivery of the project and evaluates all database design and modeling theory. At this stage students have assimilated knowledge to be able to propose an adequate model for their database. Differently, the second midterm is done after the final project delivery with the purpose to be able to assess the SQL language mastery only after being extensively used by students. Therefore, knowledge and know-how are perfectly interweaved.

Students use Moodle system with strict deadlines and significant penalties for late deliveries. The projects are orally discussed and demonstrated by the team members, complemented with the evaluation of reports and testing of submitted code. The interface is also testes and evaluated. In this way, the soft skills of this unit are achieved, in particular time management and responsible team work.

4.4.9. Bibliografia de consulta/existência obrigatória:

Recommended book

*Database System Concepts, 7th Edition.
Abraham Silberschatz, Henry F. Korth and S. Sudarshan
McGraw Hill, 2019
ISBN 9780078022159*

Slides and exercises

After each class the slides and exercises will be made available online in CLIP.

Complementary readings

*A Guide to the SQL Standard (4th Ed)
C. J. Date and Hugh Darwen
Addison-Wesley
ISBN:0-201-94426-0
The theory of Relational Databases
David Maier
Computer Science Press
ISBN:0-914894-42-0*

Mapa IV - Linguagens e Ambientes de Programação

4.4.1.1. Designação da unidade curricular:

Linguagens e Ambientes de Programação

4.4.1.1. Title of curricular unit:

Programming Languages and Environments

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:*<no answer>***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Artur Miguel de Andrade Vieira Dias- T:42; PL:28***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Fernando Pedro Reino da Silva Birra - PL:84**João Ricardo Viegas da Costa Seco - T:42**Miguel Jorge Tavares Pessoa Monteiro - PL:112***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Saber:**1 - Perceber as características e os cenários de utilização de um conjunto de linguagens de programação modernas: Caml, C, JavaScript, 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.**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.***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***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.**Skills:**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.***4.4.5. Conteúdos programáticos:***A – Ambientes de programação e execução**Interpretação. Máquinas Virtuais. Níveis de interpretação. Compilação. Ligação. Carregamento. Modelo de execução.**Portabilidade. Segurança. Interoperabilidade. Ferramentas.**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 não-locais; parametrização; closures.**C - Linguagens baseadas em objetos**JavaScript e Java. Classes e protótipos. Ocultação. 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 JavaScript.**F - Introdução à programação web**G – Escolha duma linguagem**Fatores a considerar. Cenários.*

4.4.5. Syllabus:

A – Programming and execution environments

Interpretation. Virtual Machines. Levels of interpretation. Compilation. Linking. Loading. Execution model. Portability. Security. Interoperability. Tools.

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 non-local names; parameterization; closures.

C – Object-oriented languages

JavaScript and Java. Classes and prototypes. 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 JavaScript.

F - Introduction to web programming

G – Selecting a programming language

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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 G.

Os objetivos 2 e 3, relacionados com o suporte à programação e execução, são endereçados no ponto A do programa.

O objetivo 4, de saber usar bem as linguagens estudadas, é coberto de forma direta pelo ponto E e F, 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.

4.4.6. Evidence 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 G.

The objectives 2 and 3, about programming and execution support, are addressed in the topic A.

The objective 4, on using well the studied languages, is covered directly by topics E and F, 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.

4.4.7. Metodologias de ensino (avaliação incluída):

Esta UC tem um forte caráter 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 UC são transmitidos, exemplificados e discutidos. Cerca de metade dos conceitos são convenientemente introduzidos no momento do estudo de linguagens concretas (e.g. inferência de tipos em OCaml, programação baseada em protótipos). A outra metade dos conceitos é discutida de forma independente, recorrendo às linguagens anteriormente estudadas para efeitos de ilustração (e.g. ligações, sistemas de tipos, polimorfismo de acordo com a nomenclatura de Cardelli).

Nas aulas práticas, os alunos resolvem pequenos problemas onde aplicam os conceitos e técnicas estudados. Alguns desses problemas têm caráter teórico, mas a maioria são problemas de programação.

Os projetos da UC são realizados principalmente fora das aulas.

4.4.7. Teaching methodologies (including students' assessment):

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. Many of the concepts that appear in the syllabus are conveniently presented and discussed at the time of the study of specific languages (e.g. type inference in OCaml, prototype based programming). Other concepts are better discussed in an independent manner, making use of the previously studied languages for illustrative purposes (e.g. bindings, type systems, the polymorphism nomenclature of Cardelli).

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 projects are mainly developed outside the classes.

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

Nas aulas teóricas os alunos são iniciados nos conceitos e técnicas da UC. 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 UC. 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.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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.

4.4.9. Bibliografia de consulta/existência obrigatória:

Artur Miguel Dias, Course notes including exercises.

Jason Hickey, Anil Madhavapeddy, Yaron Minsky, "Real World OCaml", O'Reilly Media, 2013.

Brian Kernighan, Dennis M. Ritchie, "The C Programming Language", 2nd ed., Prentice Hall, 1988.

Marijn Haverbeke, "Eloquent JavaScript", 2nd ed., No Starch Press, 2014.

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 - Probabilidades e Estatística D

4.4.1.1. Designação da unidade curricular:

Probabilidades e Estatística D

4.4.1.1. Title of curricular unit:

Probability and Statistics D

4.4.1.2. Sigla da área científica em que se insere:

M

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

TP:42; PL:14

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):*Maria de Fátima Varregoso Miguens - TP:42; PL:14***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:**

<sem resposta>

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

O objetivo da unidade curricular é proporcionar ao aluno uma base sólida de conhecimentos elementares de Probabilidades e Estatística que constituem uma ferramenta indispensável à tomada de decisão em situações de incerteza. Esta aquisição de conhecimentos deverá municiar os alunos de uma capacidade de aquisição futura de conceitos mais avançados que surjam no seu percurso de formação académica e/ou profissional.

No final da unidade o aluno terá adquirido competências que lhe permitam:

- Conhecer e compreender os elementos básicos da teoria e do cálculo das probabilidades
- Descrever as principais distribuições probabilísticas de variáveis discretas e contínuas e aplicá-las na descrição de fenómenos aleatórios
- Inferir sobre parâmetros populacionais com base em distribuições amostrais
- Construir modelos estatísticos que permitam estabelecer uma relação funcional entre variáveis
- Saber trabalhar com um software estatístico.

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

The aim of the course is to teach the basic concepts of probability and statistics. The students will be prepared to handle the requirements during their professional activities that concern probabilities and statistics. With regard to probabilities, the goal is for students to develop skills to formulate problems concerning the results of random observations. Students should also be able to handle statistical techniques and be familiar with a statistical software package, in order to analyse parameters of a population, e.g. to be able to use linear regression as a first approach to model real data.

4.4.5. Conteúdos programáticos:

1. Introdução à Teoria das Probabilidades
2. Variáveis aleatórias e suas distribuições de probabilidade
3. Momentos de variáveis aleatórias
4. Vetores aleatórios
5. Teorema Limite Central
6. Noções elementares de estatística
7. Estimção pontual e intervalar
8. Testes de hipóteses
9. Regressão linear simples

4.4.5. Syllabus:

1. Introduction to the theory of probability.
2. Random variables and their distributions.
3. Moments of random variables.
4. Random vectors.
5. Central limit theorem.
6. Basic notions of statistics.
7. Point and interval estimation.
8. Hypothesis testing
9. Simple linear regression

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

A componente de Probabilidades, que compreende os conteúdos programáticos 1 ao 5, destina-se a dar a conhecer as ferramentas probabilísticas fundamentais a um bom acompanhamento dos conceitos e resultados estatísticos. Cumprem-se assim os dois primeiros objetivos de aprendizagem.

Na componente de Estatística (conteúdos programáticos 6 ao 10) apresentam-se as técnicas estatísticas clássicas e de aplicação mais frequente nos problemas de inferência. Com estas matérias, pretende-se transmitir a forma de raciocínio sobre questões estatísticas, possibilitando um razoável acompanhamento e compreensão de outras técnicas mais complexas. Cumprem-se assim os dois últimos objetivos de aprendizagem.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The Probabilities component, which comprises syllabi 1 to 5, is intended to achieve understanding of the fundamental probabilistic tools for a good understanding of the concepts and results of statistics. This fulfils the first two objectives of the curricular unit.

The Statistics component (syllabi 6 to 10) presents the classic and most frequently used statistical techniques in inference problems. With this component, the students should be able to follow-up and understand other more complex techniques. This fulfils the last two objectives of the curricular unit.

4.4.7. Metodologias de ensino (avaliação incluída):

O método de ensino utilizado nesta unidade curricular pode ser resumido como se segue:

- *Os temas são introduzidos através de uma exposição oral detalhada dos conteúdos da Unidade Curricular utilizando, sempre que possível, exemplos de aplicação à matéria a ser lecionada. Pretende-se também motivar no aluno o interesse pelo estudo desta matéria. A exposição oral é feita tradicionalmente no quadro com apoio de "slides".*
- *Seguidamente são propostos e corrigidos exercícios e são tiradas dúvidas que tenham resultado do estudo dos alunos.*
- *Ao longo do semestre são realizadas provas de avaliação contínua.*

4.4.7. Teaching methodologies (including students' assessment):

The teaching method used in this course can be summarised as follows:

- *The topics are introduced through an oral presentation detailing the contents of the course using, where possible, examples of applications of the subject matter. It is also intended to motivate the student's interest in the study of this matter. The oral presentation is given traditionally using a black board, supplemented with "slides".*
- *Following this, exercises are given and corrected. Also, raised doubts by the students are clarified.*
- *Throughout the semester continuous evaluation tests are applied.*

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

As aulas são de carácter teórico-prático o que à partida permite uma ligação estreita e imediata entre os conceitos teóricos e a sua aplicabilidade.

Os alunos têm um contacto de 4h semanais com a disciplina, repartidos por dois períodos de 1h30m de aulas teórico-práticas complementados com uma hora de contacto para resolução de exercícios.

Na primeira parte da aula introduzem-se os conceitos teóricos com a ilustração de exemplos práticos, sempre que possível. Na segunda parte complementa-se a aprendizagem com a resolução de exercícios. Desta forma, os alunos têm uma visão integrada dos tópicos lecionados, fomentam o espírito crítico e o trabalho em grupo. Para que a visão integrada dos tópicos se vá mantendo ao longo do funcionamento da unidade é exigida a frequência das aulas.

O trabalho em aula é complementado com a resolução de exercícios propostos. Os alunos têm um apoio adicional no seu estudo quer com material de suporte ("sildes" e sebenta da matéria teórica, exames e testes resolvidos), quer com horários de atendimento, ambos disponíveis na página web da unidade curricular.

O cumprimento dos objetivos é avaliado de uma forma contínua ou por exame em época de recurso.

A forma contínua passa pela realização de dois testes. No primeiro teste avalia-se se os conceitos probabilísticos foram apreendidos. Garante-se assim a base para a introdução dos conceitos estatísticos. O segundo teste avalia as competências adquiridas ao nível da estatística.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

Hands-on theoretical classes are used, allowing an immediate connection between theoretical concepts and their applicability.

Students have 4 hours contact with the unit each week, divided into two periods of 1h30m each for expository classes plus 1 hour for problem solving.

In the first part of the class the theoretical concepts are introduced. The second part focused on problem solving. This way, the students have an integrated view of the topics taught, fostering critical thinking and teamwork. Class attendance is required for an integrated vision of the topics.

The class work is supplemented with practical exercises. Students have access to additional supporting material such as overhead sheets and past examination materials, and can request additional dedicated time, both available on the course's webpage.

The achievement of the objectives is assessed through continuous evaluation as well as through a final exam.

The continuous evaluation is done in two parts. The first test evaluates whether the probabilistic concepts have been learned, or in other words if the first two unit objectives have been achieved. This ensures the foundation for the introduction of the statistical concepts. The second test assesses the acquired statistics skills.

4.4.9. Bibliografia de consulta/existência obrigatória:

Guimarães, R.C. & Cabral, J.A.S. (2007), Estatística, McGraw-Hill.

Montgomery, D.C. & Runger, G.C. (2011), Applied Statistics and Probability for Engineers, John Wiley.

Paulino e Branco (2005). Exercícios de Probabilidade e Estatística. Escolar Editora.

Pedrosa, A.C. & Gama, S.M.A. (2004), Introdução Computacional à Probabilidade e Estatística, Porto Editora.

Ross, S.M. (2014). Introduction to Probability and Statistics for Engineers and Scientists, Academic Press, 5th Edition.

Mapa IV - Teoria da Computação

4.4.1.1. Designação da unidade curricular:

Teoria da Computação

4.4.1.1. Title of curricular unit:

Theory of Computation

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

António Maria Lobo César Alarcão Ravara - T:84

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Sofia Carmen Faria Maia Cavaco - PL:112

Vasco Miguel Moreira Amaral - PL:112

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Usar a lógica e a teoria de conjuntos para modelar dados e sistemas. Conhecer os fundamentos teóricos da computação, o conceito formal de algoritmo e a existência de problemas indecidíveis. Conhecer as classes de linguagens formais, os modelos computacionais associadas e a sua relação mútua. Compreender o conceito de universalidade Turing.

Modelar o espaço de estados de sistemas com conjuntos e lógica de 1.ª ordem. Distinguir conjuntos contáveis de não contáveis. Modelar sistemas com autómatos finitos (DFA e NFA). Construir um autómato dada uma expressão regular e o inverso. Construir um DFA equivalente a um NFA. Definir linguagens independentes de contexto com gramáticas. Construir analisadores LL e LR. Reconhecer a (in)decidibilidade de problemas computacionais.

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

Use logic and set theory to model data and systems. Know the theoretical foundations of computation, the formal concept of algorithm, and the existence of undecidable problems. Know the classes of formal languages, associated computational models, and the relationship between them. Understand the concept of Turing universality.

Model the state space of systems or abstract data types with set theory and first order logic. Distinguish countable from non-countable sets. Model systems using finite automata (DFA and NFA). Construct a finite automata from a regular

expression and conversely. Construct a DFA equivalent to a NFA. Define context free languages using grammars. Construct LL and LR parsers. Recognize (un)decidable computational problems.

4.4.5. Conteúdos programáticos:

Modelação com Conjuntos e Lógica: Conjuntos, Funções, Relações. Finito e Infinito, argumento diagonal de Cantor. Diferença entre função e algoritmo. Definições indutivas. Modelos de sistemas e tipos abstratos de dados.

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).

Computabilidade: Complexidade básica (P, NP). Expressividade computacional. Máquinas abstratas e níveis de interpretação. Universalidade Turing. Tese de Church-Turing. Indecidibilidade (da terminação).

4.4.5. Syllabus:

Modeling with Sets and Logic: Sets, Functions, Relations. Finite and Infinite, Cantor's diagonalization argument. Difference between function and algorithm. Inductive definitions. Models of systems and abstract data types.

Machines, Automata and Specifications: What is a model of computation? Deterministic finite automata and regular expressions. (Non-)Determinism in computation. Context Free languages and Stack machines. Parsing algorithms (LL and LR).

Computability: Basic complexity (P, NP). Expressiveness of computation models. Abstract machines and layers of interpretation. Turing universality. Church-Turing thesis. Undecidability (of termination).

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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”.

4.4.6. Evidence 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”.

4.4.7. Metodologias de ensino (avaliação incluída):

O ensino está organizado em aulas teóricas e práticas. Existem notas escritas, que seguem de perto os conteúdos das aulas teóricas, suportadas por acetatos.

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. Tipicamente 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 quatro mini-testes e dois testes. Os alunos que tenham reprovado na avaliação contínua podem fazer o exame final.

4.4.7. Teaching methodologies (including students' assessment):

The course is organized in lectures and laboratory classes. There are written lecture notes, which closely follow the presentation in the lectures.

In the laboratory classes students discuss and solve problems proposed by the instructor from a predefined list. In the lectures the instructor presents and motivates concepts and applications are discussed and exemplified, generally prompted by challenges arising in various areas of computer science. Typically, “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 three tests. Any student that fails the evaluation is admitted to the final exam.

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

Os exercícios 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.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The practical individual work assignments contribute to bind the students to a strong work rhythm 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 particular 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.

4.4.9. Bibliografia de consulta/existência obrigatória:

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 - Computação Gráfica e Interfaces**4.4.1.1. Designação da unidade curricular:**

Computação Gráfica e Interfaces

4.4.1.1. Title of curricular unit:

Computer Graphics and Interfaces

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Nuno Manuel Robalo Correia - PL:112

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Fernando Pedro Reino da Silva Birra - T:84

Rui Pedro da Silva Nóbrega - PL:112

4.4.4. Objetivos de aprendizagem (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.*

4.4.4. Intended learning outcomes (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 know the illumination models and shading techniques designed for graphics systems based on polygon's rasterization.*
- 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.*

4.4.5. Conteúdos programáticos:

Modelação
Primitivas
Grafos de cena
Projeções Geométricas
Paralela
Perspetiva
Outras transformações essenciais no pipeline gráfico
Enquadramento janela-visor
Recorte
Remoção de partes ocultas
Programação por eventos
Animação gráfica em tempo real
Introdução aos modelos de cor e de iluminação
Geração e aplicação de texturas por mapeamento:
Imagens digitais
métodos procedimentais
Interação
Modelos
Estilos
Design do ecrã
Contextos
Técnicas de interface com o utilizador
Metáforas 2D/3D
Dispositivos físicos
Controlo
Feedback
Visualização
Navegação e manipulação
Programação em WebGL

4.4.5. Syllabus:

Modeling:
Primitives
Surfaces
Scene graphs
Geometric projections:
Parallel
Perspective
Other essential transformations in the graphics pipeline:
Window-viewport transformation
Clipping
HLHSR
Event-driven programming
Computer animation in real time
Introduction to color models and illumination models
Generation and application of texture mapping:
Digital images
procedural methods
Interaction:
Models
Styles
Screen design
Contexts
User interface techniques:
2D/3D metaphors
Physical devices
Control
Feedback
Visualization
Navigation and manipulation
WebGL Programming

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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 WebGL permite a aprendizagem de ferramentas atuais com grande utilidade prática na programação numa grande diversidade de aplicações (10 e 11).

4.4.6. Evidence 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 WebGL allows learning current tools with great practical usefulness in programming of a wide variety of applications (10 and 11).

4.4.7. Metodologias de ensino (avaliação incluída):

As matérias são explicadas e ilustradas com exemplos de programação. Os alunos são desafiados em avaliar as soluções propostas e a considerar alternativas. Em geral, todos os resultados principais são mostrados com base numa variedade de meios, incluindo páginas web.

As aulas práticas estão divididas em dois grupos. As aulas de treino nas tecnologias usadas (WebGL) com aplicação dos conceitos lecionados nas aulas teóricas, e as aulas dedicadas à realização dos projetos.

Os alunos fazem três projetos em grupo que abrangem a programação de questões importantes em computação gráfica e onde projetam e implementam a interface gráfica das aplicações 2D/3D correspondentes. Os alunos devem frequentar a maioria das aulas práticas.

A nota final é calculada com os seguintes pesos: Projetos (3x) 10%, Testes (2x): 35%. Exame Final: 70%. Nota mínima na média dos testes ou no exame: 8,0 valores.

4.4.7. Teaching methodologies (including students' assessment):

Matters are explained and illustrated with programming examples. Students are invited to evaluate the proposed solutions and to consider possible alternatives, comparing them. In general, all the main results are shown using a variety of means, including web pages.

Practical classes are divided in two different types. Training sessions, meant to exercise on the usage of technology (WebGL), where concepts from lectures are also applied, and sessions dedicated to project development.

Students need to develop three 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.

Final grade is computed using the following weights: Assignments (3x): 10%, Tests (2x): 25%, Final Exam: 70%. A grade of at least 8,0 in the final exam or in the average of the tests is required for approval.

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

As aulas teóricas têm como principal objetivo transmitir conhecimento que se pretende ser sólido. A exposição dos conteúdos é uma consequência da discussão crítica que identifica as necessidades e aponta possíveis caminhos que levam à solução exposta.

Muitos dos conceitos são demonstrados com recurso a ferramentas interativas, com especial ênfase em páginas Web, construídas precisamente com as mesmas técnicas e tecnologias que as abordadas na unidade curricular.

Nas aulas práticas dedicadas ao treino em tecnologias (WebGL) aproveita-se a oportunidade para fazer a ligação com alguns dos conteúdos teóricos, tentando sempre ligar a prática à teoria. Os alunos seguem um guião que os conduz aos objetivos mínimos da sessão e têm, em cada aula, um conjunto de desafios para resolver fora das aulas, de aperfeiçoamento ou aprofundamento do conhecimento.

Nas restantes aulas práticas, cada projeto é desenvolvido ao longo de várias sessões interligando vários tópicos abordados nas aulas teóricas para a resolução de problemas que caracterizam aplicações 3D interativas, desde a disposição dos elementos no ecrã, passando pela implementação da visualização concreta pedida e da necessária interface com o utilizador.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

The main purpose of lectures is to pass knowledge in a solid way. The presentation of the syllabus contents is a consequence of a critical discussion that tries to identify the requirements and point to prospective solutions to the identified problems.

Most concepts are demonstrated through the use of interactive tools, with special emphasis on Web pages, built with similar tools and concepts as those addressed by the course.

Lab sessions dedicated to training with the base technologies (WebGL) present an opportunity to try to implement different techniques addressed in lectures. Student follow a script that leads them to the minimum objectives. Each script contains additional problems that are meant to be solved outside the class, focused either to perfecting or deepening knowledge and practice.

The remainder lab sessions, dedicated to project development throughout several sessions, serve to interconnect several aspects of knowledge and combine several tools in more complex applications. Well known and identifiable scenarios are used as problems and the students need to go through the layout of the graphical components, program the required visualization and the much needed end user interface.

4.4.9. Bibliografia de consulta/existência obrigatória:

Edward Angel and Dave Shreiner, "Interactive Computer Graphics, A Top-Down Approach with WebGL", Seventh Edition, Addison-Wesley 2015

João M. Pereira, João Brisson, António Coelho, Alfredo Ferreira e Mário Rui Gomes, "Introdução à Computação Gráfica", FCA Editora, 2018

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)

Edward Angel and Dave Shreiner, "Interactive Computer Graphics, A Top-Down Approach with WebGL", Seventh Edition, Addison-Wesley 2015

J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, "Computer Graphics - Principles and Practice", 2nd edition in C, Addison-Wesley (1996)

Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "Human-Computer Interaction", 3rd edition, Prentice Hall (2004)

Mapa IV - Engenharia de Software

4.4.1.1. Designação da unidade curricular:*Engenharia de Software***4.4.1.1. Title of curricular unit:***Software Engineering***4.4.1.2. Sigla da área científica em que se insere:**

I

4.4.1.3. Duração:*Semestral / Semester***4.4.1.4. Horas de trabalho:**

252

4.4.1.5. Horas de contacto:*T:42; PL:28***4.4.1.6. ECTS:**

9

4.4.1.7. Observações:*<sem resposta>***4.4.1.7. Observations:***<no answer>***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Vasco Miguel Moreira Amaral - T:42***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***João Baptista da Silva Araújo Junior - PL:140**Miguel Carlos Pacheco Afonso Goulão - T:42; PL:56**Miguel Jorge Tavares Pessoa Monteiro - PL:28***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):**

Esta UC de 9 ECTS tem por objetivo dotar os alunos de competências para analisar sistemas complexos existentes, e fazê-los evoluir usando metodologias e ferramentas tipicamente usadas na indústria. Assim, o aluno deve:

- *Compreender os princípios e conceitos da Engenharia de Software; requisitos de sistema; modelação usando um conjunto principal de UML; padrões e estilos arquitetónicos; DevOps; Conceção e programação de sistemas de software extensíveis; testar e evoluir o código orientado a objetos, analisar o código orientado a objetos e derivar os modelos correspondentes; Padrões de design; frameworks e APIs.*
- *Conhecer processos de engenharia de software; um conjunto principal de UML, padrões e estilos arquitetónicos; padrões de desenho; DevOps; técnicas de engenharia reversa e direta.*
- *Ser capaz de usar um sistema existente de dimensão não trivial usando técnicas sistemáticas de engenharia reversa e direta; Usar padrões, estruturas e APIs.*

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

This unit of 9 ECTS has the goal of empowering the students with knowledge and skills to analyse existing complex systems, and evolve them using methodologies and tools typically used in industry. Hence, the student must:

- *Understand the basic principles and concepts of software engineering; system requirements; modelling using a core set of UML; architectural patterns and styles; DevOps; design and programming of extensible software systems; testing and evolving object-oriented code by analysing object-oriented code and deriving the corresponding models; design patterns; frameworks and APIs.*
- *Know software engineering processes; a core set of UML; architectural patterns and styles; design patterns; DevOps; forward and reverse engineering techniques.*
- *Be able to use an existing system of non-trivial dimension using systematic reverse and forward engineering techniques; use patterns, frameworks and APIs.*

4.4.5. Conteúdos programáticos:

- *Engenharia de software: objetivos, princípios, ética, responsabilidades profissionais*
- *Gestão de projetos de software*
- *Processos de software (e.g. DevOps)*
- *Metodologias de desenvolvimento de software:*
 - *Engenharia de requisitos: tipos e técnicas de requisitos*
 - *Arquitetura de software: conceitos, padrões e estilos*
 - *Análise e desenho orientados a objetos, usando um subconjunto CORE do UML (diagramas de classes, diagramas de pacotes, máquinas de estado, diagrama de sequência)*
 - *Implementação usando orientação a objetos (e.g. Java e C#)*
 - *Verificação e validação: revisões, testes unitários, integração, aceitação e carga*
 - *Reutilização planeada e riscos*
- *Tecnologias de suporte:*
 - *Engenharia reversa (e.g. SourceMiner)*
 - *Especificação e modelação (e.g. USE, StarUML)*
 - *Implementação e evolução (e.g. Eclipse, GitHub)*
 - *Testes unitários, integração e carga (e.g. JUnit, Spock, JMeter)*
 - *Avaliação de complexidade (e.g. eclipse metrics plugin, inFusion)*
 - *Reutilização: frameworks e APIs*

4.4.5. Syllabus:

- *Software Engineering: goals, principles, ethics and professional responsibilities and liabilities*
- *Software project management*
- *Software processes (e.g. DevOps)*
- *Software Development Methodologies:*
 - *Requirements engineering: requirements types and specification techniques*
 - *Software Architecture: concepts, patterns and styles*
 - *Object-oriented analysis and design, using a core subset of UML (class diagrams, package diagrams, state machines, sequence diagrams)*
 - *Implementation using object-oriented languages (e.g. Java e C#)*
 - *Verification and validation: reviews, unit, integration and acceptance tests*
 - *Planned reuse and risks*
- *Supporting technologies:*
 - *Reverse engineering (e.g. SourceMiner)*
 - *Specification and modelling (e.g. USE, StarUML)*
 - *Implementation and evolution (e.g. Eclipse, GitHub)*
 - *Unit tests, integration, acceptance and performance testing (e.g. JUnit, Spock, JMeter)*
 - *Software complexity evaluation (e.g. eclipse metrics plugin, inFusion)*
 - *Reuse: frameworks and APIs*

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Esta unidade curricular é o ponto de partida para expor os alunos a sistemas de média ou grande dimensão, desenvolvidos por terceiros, desenvolvendo capacidades de análise e compreensão de código para posterior evolução, segundo uma abordagem DevOps. Equipas de até 6 alunos estendem projetos open source existentes (em domínios diversos tais como jogos, aplicações de negócio e ferramentas de desenvolvimento), com novas características. Para isso, os alunos aprenderão e usarão na prática um conjunto de metodologias de desenvolvimento e evolução de software, desde os requisitos à implementação e teste do software, sendo fundamental a utilização de técnicas planeadas de reutilização como a utilização de padrões e frameworks e de engenharia reversa. Os vários passos do processo de desenvolvimento (análise, especificação, desenho de solução, implementação, testes e operação) serão apoiados por tecnologias adequadas e tipicamente usadas na indústria.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

This course is the starting point for exposing students to moderate and large sized existing third-party systems, developing abilities of analysis and understanding of code for subsequent evolution, using a DevOps approach. Teams of up to 6 students will extend existing open source projects (in domains such as games, business applications and development tools). To do so, students will learn and use, in practice, a set of software development and evolution methodologies, from requirements to implementation and testing, with the fundamental contribution of planned reuse techniques, including patterns and frameworks, and reverse engineering. The several steps of the development process (analysis, specification, solution design, implementation, testing and operation) will be supported by suitable technologies typically adopted in industry.

4.4.7. Metodologias de ensino (avaliação incluída):

3 horas semanais teóricas para apresentação dos conceitos, princípios e técnicas de apoio ao desenvolvimento e evolução de software. A matéria, ilustrada com exemplos de aplicação, e suportada por diapositivos e pela utilização do ambiente de desenvolvimento no computador do docente.

2 horas semanais de aulas práticas em laboratório onde os alunos põem em prática as teorias e técnicas ensinadas das

aulas teóricas anteriores. Os alunos podem esclarecer dúvidas durante as aulas ou nos horários de atendimento. A avaliação tem duas componentes: um projeto e dois testes. O projeto dá frequência, por isso a nota mínima é 9,5 valores, e vale 30% da nota final. A média dos dois testes é 9,5 valores.

4.4.7. Teaching methodologies (including students' assessment):

3 hours a week for the presentation of concepts, principles and techniques to support the development and evolution of software. The subject, illustrated with application examples, is supported by slides and by the use of the development environment on the teacher's computer.

2 hours a week of practical laboratory classes in which students practice the theories and techniques taught in previous theoretical classes. Students can answer to questions during class or during the hours of service.

The evaluation has two components: one project and two tests. The project gives frequency, so the minimum grade is 9.5 values, and is worth 30% of the final grade. The average of the two tests is 9.5 values.

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

Nas aulas teóricas a matéria é exposta e demonstrada em exemplos. As metodologias e ferramentas são depois usadas nas aulas de laboratório para a extração das abstrações e de informação qualitativa e quantitativa, nomeadamente visualização de software (através de extração de métricas), deteção de clones, produção de diagramas UML. O passo seguinte é pedir aos alunos a modelação e implementação de novas funcionalidades para fazer evoluir o projeto. Os testes deverão incluir testes unitários, de integração, carga e aceitação. O resultado será uma nova versão do projeto de software, e ainda um relatório que documente o processo e o seu resultado. O projeto é desenvolvido primordialmente fora das aulas.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

In the theoretical classes, the students are exposed to the foundations and application examples. The methodologies and tools are then used in laboratory classes to use various techniques and tools to extract abstractions, qualitative and quantitative information, such as visualization of software (through the extraction of metrics), detection of clones, or production of UML diagrams. The next step is to ask students to model and implement new features to evolve the project. The tests should include unit tests, integration tests, performance tests and acceptance tests. The result will be a new version of the software project and also a report documenting the process and the results. The project is developed by the students mainly outside of the classroom.

4.4.9. Bibliografia de consulta/existência obrigatória:

Ian Sommerville, Software Engineering, Pearson, 10th edition, 2015

Jim Arlow and Ila Neustadt, UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition), Addison-Wesley Professional; 2 edition, July 7, 2005

Lanza M.; Marinescu R. Object-Oriented Metrics in Practice. Springer-Verlag 2006. ISBN-10 3-540-24429

L. Bass, P. Clements, R. Kazman, "Software Architecture in Practice", 3rd edition, Addison-Wesley, 2012

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995

Len Bass, Ingo Weber, Liming Zhu, DevOps: A Software Architect's Perspective (SEI Series in Software Engineering), Pearson Education, 2015

S Demeyer, T Mens, Software Evolution, Springer, 2008

Glenford J. Myers, The Art of Software Testing, Third Edition, John Wiley & Sons, Inc., 2011

Mapa IV - Inteligência Artificial

4.4.1.1. Designação da unidade curricular:

Inteligência Artificial

4.4.1.1. Title of curricular unit:

Artificial Intelligence

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:*Semestral /Semester***4.4.1.4. Horas de trabalho:**

168

4.4.1.5. Horas de contacto:*T:42; PL:28***4.4.1.6. ECTS:**

6

4.4.1.7. Observações:*<sem resposta>***4.4.1.7. Observations:***<no answer>***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***João Alexandre Carvalho Pinheiro Leite - T:84; PL:56***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Joerg Matthias Knorr - PL:84**Ricardo João Rodrigues Gonçalves - PL:84***4.4.4. Objetivos de aprendizagem (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 1.^a 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/espaço necessário para a obter.**Justificar ou selecionar soluções.**Trabalhar em equipa com a escrita de relatórios.**Capacidade de modelação abstrata.***4.4.4. Intended learning outcomes (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.*

4.4.5. Conteúdos programáticos:

- 1.Panorâmica e breve história da IA e suas aplicações
- 2.Resolução de problemas
 - 2.1Agentes e problemas de procura
 - 2.2Procura cega
 - 2.3Procura heurística em espaços de estados. Algoritmo A*.
 - 2.4Procura local e problemas de otimização
 - 2.5Agentes adaptativos
 - 2.6Problemas de Satisfação de Restrições
 - 2.7Procura com Adversários
- 3.Representação do Conhecimento e Raciocínio
 - 3.1Lógica Proposicional
 - 3.2Lógica de Predicados de 1.ª Ordem
 - 3.3Programação em Lógica
- 4.Planeamento
 - 4.1Representação de problemas de planeamento clássicos
 - 4.2Linguagens e algoritmos de planeamento
 - 4.3GRAPHPLAN
- 5.Incerteza e raciocínio probabilístico
 - 5.1Teoria das Probabilidades e Regra de Bayes
 - 5.2Redes Bayesianas
- 6.Aprendizagem Automática
 - 6.1Abordagens ao problema da aprendizagem
 - 6.2Aprendizagem concetual e indutiva
 - 6.3Redes Neurais
- 7Conclusão

Aulas PL:

*Procura informada em espaços de estados
Procura local
Procura com adversários
Programação em lógica
Planeamento
Incerteza e raciocínio probabilístico
Aprendizagem automática*

4.4.5. Syllabus:

- 1.Overview of Artificial Intelligence and its applications
- 2.Problem-solving
 - 2.1.Search agents and search problems.
 - 2.2.Blind search.
 - 2.3.Heuristic search in state-space graphs. A* algorithm.
 - 2.4.Local search and optimisation problems.
 - 2.5.Adaptative agents.
 - 2.6.Constraint Satisfaction Problems.
 - 2.7.Adversarial search.
- 3.Knowledge Representation and Reasoning
 - 3.1.Propositional logic.
 - 3.2.First-order logic.
 - 3.3.Logic programming
- 4.Planning
 - 4.1.Logical representation of classical planning problems.
 - 4.2.Planning languages and algorithms. Partial order planning.
 - 4.3.GRAPHPLAN
- 5.Uncertainty and probabilistic reasoning

5.1. *Probability theory and Bayes' rule.*

5.2. *Bayesian networks.*

6. *Machine Learning*

6.1. *Learning agents. Approaches to the learning problem.*

6.2. *Conceptual and inductive learning.*

6.3. *Neuronal networks*

7. *Conclusion*

Labs:

Informed Search

Local Search

Adversarial Search

Logic Programming

Planning

Uncertainty and Probabilistic Reasoning

Automated Learning

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Os objetivos da unidade curricular estão desenhados para dotar os alunos de conhecimentos e aptidões em tópicos centrais abordados pela Inteligência Artificial: Procura, Representação do Conhecimento e Raciocínio, Planeamento e Aprendizagem Automática. Os tópicos cobrem 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 vários trabalhos práticos de grupo e participação nas aulas práticas. As competências obtêm-se da conjugação das componentes práticas e teóricas da unidade curricular.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The objectives of the course are designed to provide students with knowledge and skills in core topics addressed by Artificial Intelligence: Search, Knowledge Representation and Reasoning, Planning and Machine Learning. The topics cover all core aspects of the ACM-IEEE CS Curriculum 2008 and Core-Tier2 in the CS Curriculum 2013 proposal, complemented by elective topics for better understanding and deepening of the subjects.

The "knowledge" is provided essentially by the theoretical component of the program, being clear its coherence, while the "know-hows" are obtained through the development of the various practical projects and participation in the practical classes. Competences are obtained by combining the practical and theoretical components of the course.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas são expostos 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 diversos temas, acompanhadas pelos docentes. Algumas das aulas práticas destinam-se à modelação e realização de exercícios selecionados de exames anteriores, em particular de redes Bayesianas e Procura com adversários. Alguns trabalhos terminam com um mini-teste para aferir o desempenho dos alunos.

Os alunos têm ao seu dispor uma página da UC que mantém informação atualizada sobre o seu funcionamento.

A avaliação é contínua e composta por uma componente sumativa que inclui a participação ativa nas aulas práticas e a realização dos trabalhos (30% da nota final), 2 testes (35% da nota final cada) durante o semestre ou um exame final (70% da nota em substituição dos testes). É necessário ter média dos testes ou nota de exame superior a 9,5 valores.

4.4.7. Teaching methodologies (including students' assessment):

In the theoretical classes, the essential points of the subject are presented orally and should be complemented with the reading of the advised manual. The laboratory classes are intended for the presentation and realization of practical work on various topics, accompanied by teachers. Some of the practical classes are designed for modeling and performing selected exercises from previous exams, in particular from Bayesian networks and Quest with opponents. Some assignments end with a quiz to assess student performance.

Students have at their disposal a web page that keeps up-to-date information on how it works.

The assessment is continuous and consists of a summative component that includes active participation in practical classes and the completion of the assignments (30% of the final grade), 2 tests (35% of the final grade each) during the semester or a final exam. (70% of grade in replacement of tests). It is necessary to have a test average or exam grade higher than 9.5.

4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos 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 vários trabalhos práticos. No primeiro deles são utilizados e modificados diversos algoritmos de procura cega, informada e local. No segundo, é implementado o algoritmo de simulated annealing. No terceiro são resolvidos exercícios de procura com adversários. No quarto, são resolvidos vários problemas usando programação por conjuntos de resposta. No quinto trabalho são modelados diversos problemas de planeamento usando PDDL. No sexto trabalho são resolvidos exercícios com redes bayesianas. No sétimo trabalho é explorada a utilização de redes neuronais para problemas de classificação.

Os momentos de avaliação sumativa também se encontram organizados para reforçar a aprendizagem, nomeadamente terminando alguns dos trabalhos práticos com um quiz.

Algumas das atividades das aulas práticas (e.g. Procura com adversários, planeamento e redes bayesianas) funcionam como atividades preparatórias para os testes, nomeadamente permitindo o treino de modelação e resolução de problemas.

As competências são obtidas essencialmente pela realização dos trabalhos em equipa e sua avaliação experimental. 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.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

All topics mentioned in the “Knowledge” objectives are dealt with in great detail in the theoretical classes of the curricular unit, being further elaborated in the chapters of the adopted text book. Topics are presented and illustrated with concrete problems, with an in-depth analysis of the characteristics, properties, advantages and disadvantages of the approaches and their algorithms. Videos and tools exploring the real applications of Artificial Intelligence and the operation of the studied algorithms are also shown. These topics are studied at a level of detail that allow students to model and solve small problems, grasping the difficulty of solving similar problems in realistic applications.

The “Know how” component is obtained by doing several practical projects. In the first one, several uninformed, informed and local search algorithms are used and modified. In the second, the simulated annealing algorithm is implemented. In the third, search exercises with opponents are solved. In the fourth, several problems are solved using Answer-Set Programming. In the fifth, several planning problems are modeled using PDDL. In the sixth, students solve exercises with Bayesian networks. In the seventh project, the use of neural networks for classification problems is explored.

Summative assessment moments are also organized to reinforce learning, including finishing some of the practical work with a quiz.

Some of the activities in the hands-on classes (eg adversarial search, planning and Bayesian networks) function as preparation activities for the tests, notably by enabling training in modeling and problem solving.

Competences are obtained essentially by carrying out the teamwork and its experimental assessment. The tests and exams are designed to assess the student's theoretical knowledge as well as their ability to model new problems, discuss and evaluate their solution.

4.4.9. Bibliografia de consulta/existência obrigatória:

The adopted book is [Stuart Russell & Peter Norvig, 2010].

Some topics are covered only in [Costa & Simões, 2008].

The remaining books are optional readings.

Adopted

*Stuart Russel and Peter Norvig. Artificial Intelligence. A Modern Approach, 3rd edition. Prentice-Hall, Inc., 2010.
E. Costa e A. Simões. Inteligência Artificial. Fundamentos e Aplicações. 2nd Edition, FCA, 2008.
L. Sterling and E. Shapiro. The Art of Prolog (2nd Ed.). MIT Press, 1994.*

Complementary

*Matt Ginsberg. Essentials of Artificial Intelligence. Morgan Kaufmann, 1994
Nils J. Nilson. Artificial Intelligence: A new synthesis. Morgan Kaufmann, 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 - Redes de Computadores

4.4.1.1. Designação da unidade curricular:

Redes de Computadores

4.4.1.1. Title of curricular unit:

Computer Networks

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

José Augusto Legatheaux Martins - T:84; PL:28

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Vítor Manuel Alves Duarte - PL:84

Sérgio Marco Duarte - PL:112

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Conhecer: redes de computadores: terminologia, organização, serviços, componentes e princípios estruturantes; a 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: 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, e simulação de protocolos.

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

Learning goals and 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

Know-how: 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; and protocol simulation.

4.4.5. Conteúdos programáticos:

Redes de computadores: terminologia, organização, serviços, componentes e princípios estruturantes (o que é uma rede, estruturação de aplicações e protocolos aplicacionais: requisitos das aplicações e transporte, canais, multiplexagem estatística, store & forward, comutação de pacotes, estrutura em camadas, ...)

Transmissão fiável de informação: requisitos e técnicas específicas (Stop & Wait, Qualidade de serviço, TCP, Difusão e FEC - Forward Error Correction)

Nomes e endereços (camadas de designação, DNS, endereços IP, endereços MAC). Difusão de informação (HTTP, desempenho e escalabilidade, ...). Redes de distribuição de conteúdos e vídeo sobre HTTP

Encaminhamento: princípios, endereçamento e algoritmos mais comuns. Redes e protocolos TCP/IP: DHCP, ARP, NAT, IP, encaminhamento em IP.

4.4.5. Syllabus:

Computer networks: terminology, organization, services, components and structuring principles (what is a network, application structuring and application protocols: applications and transport requirements, channels, statistical multiplexing, store & forward, packet switching, layered structure, ...)

Reliable transmission of information: specific requirements and techniques (Stop & Wait, Quality of service, TCP, Diffusion and FEC - Forward Error Correction)

Names and addresses (designation layers, DNS, IP addresses, MAC addresses). Diffusion of information (HTTP, performance and scalability, ...). HTTP and video content distribution networks

Routing: principles, addressing and most common algorithms. TCP / IP networks and protocols: DHCP, ARP, NAT, IP, IP forwarding.

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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 protocolos e aplicações distribuídas, usando as interfaces de rede de nível transporte, devem ser exercitadas.

Aspetos 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, nomes e endereços, o encaminhamento e sua análise, e o funcionamento dos canais partilhados são os temas de base que se procura que o aluno domine. Estes tópicos são de forma geral reconhecidos como os aspetos fundamentais das redes de computadores que os alunos de Engenharia Informática devem dominar.

4.4.6. Evidence 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 protocols and distributed applications using transport-level network interface functions is exercised.

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, names and addresses, routing and its analysis, and shared channels and medium control access. These are recognized as the main computer network related topics for Computer Science students.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas expõe-se e discute-se o programa da UC. 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.

A avaliação é realizada através de dois testes intermédios (com o peso de 35% cada na nota final) e ainda vários trabalhos laboratoriais de grupo (com o peso total de 30% na nota final).

4.4.7. Teaching methodologies (including students' assessment):

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.

Grading is based on two intermediate tests (with the weight of 35% each in the final grade) as well as several group laboratory projects (with the total weight of 30% in the final grade).

4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos 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.

Aspetos 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, nomes e endereços, o encaminhamento e sua análise, e o funcionamento dos canais partilhados são os temas de base que se procura que o aluno domine. Estes tópicos são de forma geral reconhecidos como os aspetos fundamentais das redes de computadores que os alunos de Engenharia Informática devem dominar.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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 distributed applications using transport-level network interface functions is exercised.

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, names and addresses, routing and its analysis, and shared channels and medium control access. These are recognized as the main computer network related topics for Computer Science students

4.4.9. Bibliografia de consulta/existência obrigatória:

José Legatheaux Marins, "Fundamentos de Redes de Computadores - Ilustrado com base na Internet e nos Protocolos TCP/IP," FCT da Universidade Nova de Lisboa, 2018

James F. Kurose and Keith W. Ross, "Computer Networking - A Top-Down Approach," Addison Wesley — Pearson Education, 2012, ISBN-10: 0136079679, 6th Edition

Mapa IV - Análise e Desenho de Algoritmos

4.4.1.1. Designação da unidade curricular:

Análise e Desenho de Algoritmos

4.4.1.1. Title of curricular unit:

Design and Analysis of Algorithms

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Margarida Paula Neves Mamede - T:84; PL:56

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Bernardo Parente Coutinho Fernandes Toninho - PL:56

Fernanda Maria Barquinha Tavares Vieira Barbosa - PL:112

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Saber

-Definir e identificar três técnicas de desenho de algoritmos: estratégias greedy, programação dinâmica e transformação-e-conquista.

-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 algumas classes de complexidade e compreender alguns problemas em aberto.

Saber Fazer

-Conceber e analisar um algoritmo aplicando programação dinâmica.

-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.

-Provar que um problema é NP-completo.

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

Knowledge

Define and identify three algorithm design techniques: greedy strategies, dynamic programming and transform-and-conquer.

Know the fundamental graph algorithms, the required abstract data types and the data structures used to implement them efficiently.

Understand amortized analysis.

Define some complexity classes and understand some open problems.

Application

Design and analyse a dynamic programming algorithm.

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.

Make NP-complete proofs.

4.4.5. Conteúdos programáticos:

(1) Programação dinâmica.

(2) Introdução ao estudo de grafos. Definições fundamentais. Tipos de dados abstratos de grafo não orientado e grafo orientado. Implementações de grafos.

(3) Algoritmos elementares de grafos. Percursos em profundidade e em largura. Ordenação topológica.

(4) Árvores mínimas de cobertura. Algoritmo de Kruskal. Tipo abstrato de dados partição.

(5) Complexidade amortizada. Método do potencial.

(6) Algoritmo de Prim. Tipo abstrato de dados fila com prioridade adaptável.

(7) Caminhos mais curtos. Algoritmos de Dijkstra, Bellman-Ford e Floyd-Warshall.

(8) Redes de fluxos. Fluxos máximos. Método de Ford-Fulkerson. Algoritmo de Edmonds-Karp. Emparelhamentos máximos em grafos bipartidos. Cortes mínimos.

(9) Introdução à Teoria da Complexidade. As classes P, NP, PSPACE e EXPTIME. Os sufixos difícil e completo. Redução de problemas. Alguns problemas em aberto.

4.4.5. Syllabus:

- (1) *Dynamic programming.*
- (2) *Introduction to the study of graphs. Fundamental definitions. The abstract data types undirected graph and directed graph. Implementations of graphs.*
- (3) *Elementary graph algorithms. Depth-first and breadth-first traversals. Topological sorting.*
- (4) *Minimum spanning trees. Kruskal's algorithm. The disjoint sets abstract data type.*
- (5) *Amortized analysis. The potential method.*
- (6) *Prim's algorithm. The adaptable priority queue abstract data type.*
- (7) *Shortest paths. The algorithms of Dijkstra, Bellman-Ford, and Floyd-Warshall.*
- (8) *Flow networks. Maximum flows. The Ford-Fulkerson method. The Edmonds-Karp algorithm. Maximum bipartite matchings. Minimum cuts.*
- (9) *Introduction to the Theory of Complexity. The classes P, NP, PSPACE, and EXPTIME. The suffixes hard and complete. Problem reductions. Some open problems.*

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Os objetivos podem ser resumidos assim.

- (a) *Conhecer os algoritmos fundamentais de grafos e as suas implementações eficientes: Os pontos 3, 4, 6, 7 e 8 cobrem os algoritmos, os novos TADs e as novas estruturas de dados que permitem implementações eficientes. (Novo é não ser estudado na UC Algoritmos e Estruturas de Dados.)*
- (b) *Calcular complexidades amortizadas, avaliar soluções e efetuar escolhas fundamentadas: A complexidade amortizada, introduzida e exemplificada no ponto 5, é necessária para se poderem analisar (avaliar) todos os algoritmos apresentados.*
- (c) *Aplicar programação dinâmica e identificar as técnicas de desenho de algoritmos greedy e transformação-e-conquista: A primeira técnica é estudada no ponto 1 e volta a ser aplicada no ponto 7; as outras são explicadas nos pontos 4, 6, 7 e 8, quando são usadas para desenhar alguns dos algoritmos estudados.*
- (d) *Definir algumas classes de complexidade e compreender alguns problemas em aberto: coberto no ponto 9.*

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The unit objectives can be summarized in the following way.

- (a) *To know the fundamental graph algorithms and their efficient implementations: Syllabus topics 3, 4, 6, 7, and 8 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 Algorithms and Data Structures unit).*
- (b) *Perform amortized analyses, evaluate solutions and justify choices: Amortized analysis, which is introduced and exemplified in topic 5, is needed to be able to analyse (evaluate) all presented algorithms.*
- (c) *Apply dynamic programming and identify the algorithm design techniques greedy and transform-and-conquer: The first technique is studied in topic 1 and applied again in topic 7; the others are explained in topics 4, 6, 7 and 8, when they are used to design some of the studied algorithms.*
- (d) *Define some complexity classes and understand some open problems: this is covered in topic 9.*

4.4.7. 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 três 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.

Em caso de reprovação, se a média das notas dos trabalhos não for inferior a 7, os alunos podem realizar um exame final. Os testes e o exame são com consulta (a papel).

*Condição para obter aprovação: $NotaP \geq 7.0$ e $NotaT \geq 9.5$ e $NotaFinal = 0.4 * NotaP + 0.6 * NotaT \geq 10$, onde: $NotaP$ é a média das notas dos trabalhos; $NotaT$ é a média das notas dos testes ou a nota do exame; $NotaFinal$ é a nota final.*

4.4.7. Teaching methodologies (including students' assessment):

There are three hours of lectures and a lab session each week. In the laboratory, students design, analyse and implement algorithms.

Assessment comprises three 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.

In case of failure, if the mean of the project grades is not less than 7 (out of 20), students can do a final exam. The tests and the exam are open-book.

*Condition to succeed: Pgrade ≥ 7.0 and Tgrade ≥ 9.5 and Fgrade = $0.4 * Pgrade + 0.6 * Tgrade \geq 10$, where: Pgrade is the mean of the project grades; Tgrade is the mean of the test grades or the exam grade; Fgrade is the final grade.*

4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos 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.). Regra geral, para resolver todos os problemas, é necessário aplicar várias vezes a técnica da programação dinâmica e adaptar todos os algoritmos de grafos apresentados 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 e, em muitos casos, não 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 e a forma de provar que um problema é NP-completo.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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. In general, to solve all problems, students must apply the dynamic programming technique several times and have to adapt every graph algorithm presented in the lectures. 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 and they are often unavailable. 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 amortised analysis and the way NP-completeness proofs are made.

4.4.9. Bibliografia de consulta/existência obrigatória:

Main References

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

Jon Kleinberg and Éva Tardos. Algorithm Design. Addison-Wesley, 2005.

Complementary References

Anany Levitin. Introduction to The Design and Analysis of Algorithms (3rd ed.). Addison-Wesley, 2012.

S. Sridhar. Design and Analysis of Algorithms. Oxford University Press, 2014.

Dexter Kozen. The Design and Analysis of Algorithms. Springer, 1992.

Michael R. Garey and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, 1979.

Steven S. Skiena. The Algorithm Design Manual (2nd ed.). Springer, 2008.

Steven S. Skiena and Miguel A. Revilla. Programming Challenges: The Programming Contest Training Manual. Springer, 2003.

Mapa IV - Atividade de Desenvolvimento Curricular**4.4.1.1. Designação da unidade curricular:**

Atividade de Desenvolvimento Curricular

4.4.1.1. Title of curricular unit:

Curricular Development Activity

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

336

4.4.1.5. Horas de contacto:

TP:28; PL:28; S:28; OT:10

4.4.1.6. ECTS:

12

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Pedro Abílio Duarte de Medeiros - TP:28; PL:28; OT:2

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

Carlos Augusto Isaac Piló Viegas Damásio - TP:28; PL:28

Henrique João Lopes Domingos - TP:28; PL:28

João Carlos Antunes Leitão - TP:28; PL:28

João Ricardo Viegas da Costa Seco - TP:28

José Augusto Legatheaux Martins - S:28

Todos os docentes da área científica do ciclo de estudos.

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Oferecida no fecho do curso, esta UC concede aos estudantes experiência integrada, em contexto aberto, de atividade técnica aplicada, adequada ao seu nível de conhecimentos. O contexto variável, por opção dos estudantes, poderá ser

equipa de investigação ou laboratório ou ambiente empresarial/industrial. Esta UC integra Programa de introdução à Prática Profissional (PIPP) e Programa de Introdução à Investigação Científica (PIIC) do “Perfil Curricular FCT” 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.

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

Offered in the end of the program, 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, integrating the offer UROP and UPOP included in “FCT 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.

4.4.5. 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 do atividade. Integração prática de conhecimentos de base adquiridos em várias unidades curriculares de carácter mais fundamental deste 1.º ciclo, 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).

4.4.5. 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 this 1st cycle courses, 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).

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Os conteúdos programáticos endereçam precisamente os conteúdos 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).

4.4.6. Evidence 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 an 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).

4.4.7. 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 UC do curso, 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.

4.4.7. Teaching methodologies (including students' assessment):

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 uniformization 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.

4.4.8. 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, proatividade 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 incorpora as atividades PIPP e PIIC do "Perfil Curricular FCT" de forma flexível e a uma muito maior profundidade, permitindo ao estudante ao longo de 20 semanas ter experiência de introdução á investigação como em PIIC na vertente Investigação de ADC, uma experiência de atividade em empresa na vertente Estágio, e uma experiência de atividade PIPP e uma experiência mais intensa de introdução de projeto e inovação/empreendedorismo na vertente Projeto.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

Results from the description given above, and of considering the special nature the Activity for Curricular Development (ACD), 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 an 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). 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, pro-activity 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 integrates the UROP and UPOP activities of "FCT Curricular Profile" in a flexible and extended (in time) way, allowing for instance a student to develop some introduction to research (like in UROP) in the ACD Research variant, an experience of activity in a company in ACD Internship variant experience or a more intensive involvement in innovative product development and entrepreneurship in ACD/Project, like in UPOP.

4.4.9. Bibliografia de consulta/existência obrigatória:

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 - Sistemas Distribuídos

4.4.1.1. Designação da unidade curricular:

Sistemas Distribuídos

4.4.1.1. Title of curricular unit:

Distributed Systems

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:42; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Nuno Manuel Ribeiro Preguiça - T:84; PL:28

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

João Carlos Antunes Leitão - PL:84

Sérgio Marco Duarte - PL:112

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

UC obrigatória e introduz os modelos, métodos e técnicas básicas para o desenvolvimento de SD seguros e tolerantes a falhas. Como pré-requisito, os estudantes devem possuir conhecimentos de algoritmos, programação e redes de computadores.

Conhecimentos

**Conhecer os problemas e desafios inerentes ao desenho de SD;*

**Compreender as arquiteturas de SD;*

**Conhecer os modelos e sistemas de comunicação direta e indireta;*

**Compreender os algoritmos para estabelecimento de canais seguros;*

**Conhecer as alternativas para a nomeação, incluindo serviços de nomes e diretório;*

**Compreender os mecanismos de ordenação e registo de causalidade de eventos;*

**Compreender os algoritmos de caching e replicação de dados e modelos de consistência associados.*

Aptidões e competências

**Desenhar e implementar um SD seguro para um problema não trivial;*

**Utilizar sistemas de comunicação normalizados para a programação de SD;*

**Utilizar replicação para melhorar desempenho e tolerar falhas.*

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

This is a mandatory course on distributed systems. It provides the basic knowledge on the models, methods and techniques for developing secure distributed systems. As prerequisites students should have previous acquaintance with algorithms, programming, and computer networks.

Knowledge

- * *Understand the problems and challenges for the design of dist. sys.;*
- * *Understand the architectures of dist. sys.;*
- * *Know the models and systems for direct and indirect communication;*
- * *Understand the algorithms for establishing secure channels;*
- * *Know the alternatives for naming: name and directory services;*
- * *Understand the mechanisms for ordering and tracking causality of events;*
- * *Understand the caching and replication algorithms and associated consistency models.*

Know-how

- * *To design a secure dist. sys. for solving a non-trivial problem.*
- * *To program dist. sys. leveraging standard communication sub-systems;*
- * *To use of replication as a technique for improving fault tolerance and performance.*

4.4.5. Conteúdos programáticos:

1. Introdução

2. Arquiteturas e modelos

2.1 Arquiteturas: cliente/servidor, p2p e variantes

2.2 Modelos de sistema

3. Comunicação direta

3.1 Sistemas base: TCP, UDP, IP multicast, HTTP

3.2 Sistemas assíncronos: HTTP assíncrono, web sockets

4. Invocação remota

4.1 Modelo

4.2 Interfaces e representação de dados

4.3 Protocolos

4.4 Ligação e concorrência no servidor

5. Interação cliente/servidor na web

5.1 SOAP

5.2 REST

6. Segurança

6.1 Técnicas criptográficas

6.2 Autenticação, certificados e canais seguros

6.3 TLS e O.Auth

7. Tempo e ordenação de eventos

7.1 Relógios físicos

7.2 Relógios lógicos, vetoriais e vetores versão

8. Replicação e consistência

8.1 Caching

8.2 Replicação forte e fraca

9. Comunicação indireta

9.1 Comunicação em grupo

9.2 Sistemas publish/subscribe e message queues

9.3 Tolerância a falhas

10. Nomes

10.1 Conceitos

10.2 Serviços de nomes e diretório

4.4.5. Syllabus:

1. Introduction

2. Architectures

2.1 Architectures: client/server, p2p and variants

2.2 System models

3. Direct communication

3.1 Base systems: TCP, UDP, IP multicast, HTTP

3.2 Asynchronous systems: HTTP asynchronous, web sockets

4. Remote invocation

4.1 Model

4.2 Interfaces and data representation

4.3 Protocols

4.4 Binding and server concurrency

5. Client/server invocation in the web

5.1 SOAP

5.2 REST

6. Security

6.1 Cryptography

6.2 Authentication, certificates and secure channels

6.3 TLS and O.Auth

7. Time and ordering of events

7.1 Limitations of physical clocks

7.2 Logical clocks, vector clocks and version vectors

8. Replication and consistency

8.1 Caching

8.2 Replication: strong: primary-backup; weak: epidemic replication and Coda

9. Indirect communication

9.1 Group communication: properties

9.2 Pub/sub and message queues

9.3 Providing fault tolerance

10. Naming

10.1 Concepts

10.2 Name and directory services

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

A UC visa fornecer um conjunto de conhecimentos e competências que permitam desenvolver sistemas distribuídos seguros 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 de sistema distribuídos. De seguida, estudam-se as técnicas de comunicação base usadas na criação de sistemas distribuídos e os protocolos para criação de canais seguros. O programa da UC 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 são colocados em prática desenvolvendo de forma incremental, um sistema distribuído com clientes e servidores desenvolvidos de raiz e interage com serviços disponíveis na Internet, permitindo aos estudantes ganhar experiência no desenvolvimento deste tipo de sistemas.

4.4.6. Evidence 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 secure 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.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas expõe-se e discute-se o programa da UC, apresentando os problemas e discutindo as respetivas soluções recorrendo a exemplos concretos da utilização de sistemas distribuídos.

As aulas práticas acompanham os dois trabalhos da UC, fornecendo aos alunos experiência no desenvolvimento de sistemas distribuídos seguros usando as técnicas estudadas. Os dois trabalhos 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 trabalhos, e presta-se apoio à sua realização.

A avaliação será baseada nos seguintes componentes: dois testes (30% cada); dois trabalhos práticos de programação, resolvidos em grupo (20% cada).

4.4.7. Teaching methodologies (including students' assessment):

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 (30% each) and two programming projects solved in team (20% each).

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

As metodologias de ensino têm como objetivo permitir conhecer os modelos e técnicas usadas no desenvolvimento de sistemas distribuídos seguros, assim como permitir aos estudantes colocar estas 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.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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 is consolidated and exercised by the presentation of examples and the execution of two programming projects.

4.4.9. Bibliografia de consulta/existência obrigatória:

Distributed Systems: Concepts and Design

George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair

Publisher: Addison Wesley; 5th edition

ISBN-13: 978-0132143011

Mapa IV - Programa de Introdução à Investigação Científica em Engenharia Informática

4.4.1.1. Designação da unidade curricular:

Programa de Introdução à Investigação Científica em Engenharia Informática

4.4.1.1. Title of curricular unit:

Undergraduate Research Opportunity Program

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Trimestral / Trimester

4.4.1.4. Horas de trabalho:

80

4.4.1.5. Horas de contacto:

OT:7

4.4.1.6. ECTS:

3

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Pedro Abílio Duarte de Medeiros - OT:7

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

<sem resposta>

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Esta unidade curricular tem como objetivo proporcionar ao estudante um primeiro contacto com a investigação científica. Cada estudante tem de escolher uma das propostas de investigação feitas pelos orientadores.

A solução concreta do problema, permite ao estudante um primeiro contacto com a investigação e os seus métodos de trabalho.

O trabalho autónomo do estudante, supervisionado pelo professor, permite também aos estudantes aplicarem os conhecimentos previamente adquiridos à solução de um problema concreto, e assim aprenderem através da realização concreta.

A escrita de um relatório final com uma apresentação resumida do problema, a forma como este foi resolvido e os resultados obtidos, permite ao estudante praticar e melhorar através de aconselhamento as suas competências de escrita, de síntese, de captação da essência das coisas, assim como da descrição de problemas e soluções a um nível de abstração adequado.

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

This course provides the students with a first contact with the practice of scientific research. Each student must choose one of the available research proposals made by the advisors.

Its main goal is to give the students a chance to apply the knowledge they acquired throughout their studies to solving concrete research problems, and this way learn by doing.

It comprises the solution of a research problem, and write-up of a report describing the problem, the approach, and the results of the work.

The concrete research problem to be addressed will be defined by the supervisor's proposal.

The solution of concrete research problems under the guidance of the proposal's supervisor will enable students to get some exposure to research topics and learn the basics of research methods.

4.4.5. Conteúdos programáticos:

De acordo com o problema de investigação proposto.

4.4.5. Syllabus:

According to the research problem proposal.

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:

Tendo em consideração os objetivos da disciplina, o seu programa e os métodos de trabalho são os adequados a uma aprendizagem baseada na realização concreta.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The course program and work methods and a learning method mostly based on learning by doing are coherent with the course goals.

4.4.7. Metodologias de ensino (avaliação incluída):

Esta unidade curricular permite aos estudantes um primeiro contacto com a prática da investigação científica. O seu objetivo principal é dar aos estudantes a oportunidade de aplicarem os conhecimentos previamente adquiridos na solução de um problema e por essa forma aprender fazendo.

O trabalho compreende o estudo e solução do problema, a escrita de um relatório descrevendo o problema e o método de solução adotado, assim como os resultados obtidos. O problema a resolver é o enunciado pela proposta de investigação feita pelo orientador proponente.

Este trabalho, conduzido sob supervisão do orientador, permitirá ao estudante um primeiro contacto com tópicos e métodos de investigação.

4.4.7. Teaching methodologies (including students' assessment):

This course provides the students with a first contact with the practice of scientific research. Its main goal is to give the students a chance to apply the knowledge they acquired throughout their studies to solving concrete research problems, and this way learn by doing.

It comprises the solution of a research problem, and write-up of a report describing the problem, the approach, and the results of the work. The concrete research problem to be addressed will be defined by the supervisor's proposal.

The solution of concrete research problems under the guidance of the proposal's supervisor will enable students to get some exposure to research topics and learn the basics of research methods.

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

Um método de aprendizagem baseado em realização concreta e solução de problemas: compreensão do problema, pesquisa do trabalho relacionado, proposta de solução, implementação e avaliação da mesma, e escrita de um relatório final é coerente com os objetivos da unidade curricular.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:

A learning method mostly based on learning by doing: problem understanding, research of related work, solution proposal, evaluation of the solution and writing of a research report, is coherent with the course goals.

4.4.9. Bibliografia de consulta/existência obrigatória:

Esta está dependente da proposta de investigação selecionada pelo estudante./According to the research problem proposal.

Mapa IV - Programa de Introdução à Prática Profissional em Engenharia Informática

4.4.1.1. Designação da unidade curricular:

Programa de Introdução à Prática Profissional em Engenharia Informática

4.4.1.1. Title of curricular unit:

Undergraduate Practice Opportunities Program

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Trimestral / Trimester

4.4.1.4. Horas de trabalho:

80

4.4.1.5. Horas de contacto:

OT:7

4.4.1.6. ECTS:

3

4.4.1.7. Observações:

<sem resposta>

4.4.1.7. Observations:

<no answer>

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):

Pedro Abílio Duarte de Medeiros - OT:7

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:

<sem resposta>

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

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.

4.4.4. Intended learning outcomes (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.

4.4.5. Conteúdos programáticos:

Os conteúdos programáticos específicos dependem do projeto concreto escolhido pelo estudante no programa.

4.4.5. Syllabus:

The concrete syllabus depends on the specific project chosen by the student in the program.

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos 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 permite aos estudantes interessados a participação em atividades em ambiente empresarial. 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, resulta necessariamente um contacto e conhecimento das práticas de trabalho de engenharia da empresa.

As técnicas de comunicação são exigidas, e testadas, para a avaliação final.

Na opção Projeto de Engenharia e Inovação, as mesmas competências são obtidas da realização de tutoriais e no desenvolvimento de uma aplicação. Nesta vertente são ainda exercitadas as competências de planeamento de projeto e gestão de pequenas equipas de desenvolvimento.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:

The formation of the list of offerings within the UPOP programme is performed by the course's scientific committee and will in fact allow interested students to participate in activities in a business environment. It is ensured that the student's activities, supervised by the advisor, are integrated in company teams. Contact with the team and advisor in the company necessarily results in contact and knowledge of the company's engineering work practices. Communication techniques are required, and tested, for the final assessment.

In the Engineering Design and Innovation option, the same skills are obtained from tutorials and application development. In this aspect, the project planning and management skills of small development teams are also exercised.

4.4.7. 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. 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.

4.4.7. Teaching methodologies (including students' assessment):

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.

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.

4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos 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.

4.4.8. Evidence of the coherence between the teaching methodologies and the 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.

4.4.9. Bibliografia de consulta/existência obrigatória:

A bibliografia depende do projeto concreto escolhido pelo estudante no programa. / The bibliography depends on the specific project chosen by the student in the program.

4.5. Metodologias de ensino e aprendizagem

4.5.1. Adequação das metodologias de ensino e aprendizagem aos objetivos de aprendizagem (conhecimentos, aptidões e competências) definidos para o ciclo de estudos:

Na maioria das UC de Informática é combinada uma vertente teórica, centrada nos aspetos conceituais da UC, com uma vertente prática na qual os estudantes criam programas informáticos que ilustram os conceitos e técnicas expostos. A combinação das duas vertentes permite não só adquirir os conhecimentos definidos nos objetivos de aprendizagem das UC, compreendendo os fundamentos e os métodos da Engenharia Informática, mas também competências que permitem colocar estes conhecimentos em prática para a resolução de problemas, lidando com diferentes facetas dos sistemas informáticos. A vertente prática é particularmente relevante num conjunto de UC de 9 ECTS, em que os trabalhos colocam aos estudantes desafios que exigem que os alunos lidem com problemas de alguma complexidade de forma autónoma. No 6.º semestre a UC Atividade de Desenvolvimento Curricular, em todas as suas vertentes, também coloca a necessidade de resolução de um problema não trivial de forma autónoma e eventualmente inovadora.

4.5.1. Evidence of the teaching and learning methodologies coherence with the intended learning outcomes of the study programme:

In most Computer Science and Engineering UC, a theoretical approach, centered on the conceptual aspects of the subject, is combined with a practical one in which students create computer programs that illustrate the concepts and techniques exposed. The combination of the two aspects allows the students not only to acquire the knowledge defined in the learning objectives of the UC, understanding the fundamentals and methods of CSE, but also the skills that allow them putting this knowledge into practice for problem solving, when dealing with different facets of computer systems. The practical part is particularly relevant in a set of 9 ECTS UC, where assigned tasks require students to deal with problems of some complexity on their own. In the 6th semester the Curriculum Development Activity, in all its options, also raises the need to solve a non-trivial problem in an autonomous and sometimes innovative way.

4.5.2. Forma de verificação de que a carga média de trabalho que será necessária aos estudantes corresponde ao estimado em ECTS:

Para cálculo dos créditos ECTS das UC 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 desde o início do processo de Bolonha, e quando necessário reajustado pelos docentes e comissão científica do curso; também são considerados os resultados de inquéritos periódicos na FCT NOVA. Na presente proposta a creditação das UC procura equilibrar o esforço do estudante entre os vários semestres do curso. Na presente proposta, a abordagem já usada anteriormente no MIEI de ter apenas 4 unidades curriculares simultâneas nos quatro primeiros semestres do curso foi estendida a todo o curso; este número contribui para o aumento da taxa de sucesso. De acordo com as recomendações gerais da FCT NOVA, todas as unidades curriculares são medidas em múltiplos de 3 ECTS, por questões de modularidade e flexibilidade.

4.5.2. Means to verify that the required students' average workload corresponds the estimated in ECTS.:

Our calculation of ECTS credits of curricular units is based on the correspondence of 1 ECTS unit to 28 working hours. After the start of the Bologna process, 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; periodic student surveys made by FCT NOVA are also used. When preparing this proposal, the ECTS for all CU was again reconsidered, with a special concern to better balance the students' work load along the several terms of the program. In the present proposal, the approach already used in MIEI of having only 4 simultaneous curricular units in the first four semesters of the programme was extended to the whole course; This number contributes to increase the student's progression. According to the general recommendations of FCT NOVA, all CU are measured in multiples of 3 ECTS units, for the sake of modularity and flexibility.

4.5.3. Formas de garantia de que a avaliação da aprendizagem dos estudantes será feita em função dos objetivos de aprendizagem da unidade curricular:

A Comissão Científica (CC) do curso em colaboração com os docentes monitoriza o ajuste que existe entre os métodos de avaliação das UC (detalhados nas fichas das UC disponíveis no sistema académico CLIP) e os objetivos de aprendizagem. Um dos aspetos do acompanhamento feito pelo CC é a elaboração de um calendário de avaliações em que são harmonizadas as datas de realização de testes e de entrega de trabalhos; as opiniões dos estudantes também são consideradas, veiculadas sobretudo pelos estudantes que fazem parte da Comissão Pedagógica do curso. O sistema de gestão de qualidade que está implementado na escola, em que são contabilizadas as opiniões dos docentes, estudantes, regentes e responsáveis das UC e presidente de Departamento contribui também para avaliar os aspetos referidos.

4.5.3. Means of ensuring that the students assessment methodologies are adequate to the intended learning outcomes:

The Scientific Committee (CC) of the programme, in collaboration with the teaching staff, monitors the match between the courses evaluation methods (available in the CLIP academic system) and the learning objectives. One of the aspects of the CC's monitoring is the preparation of a schedule of evaluations harmonizing the dates of tests and submission of practical assignments; Students' opinions are also considered, mainly conveyed by students who are part of the Pedagogical Committee of the programme.

The school quality management system, which collects opinions of teachers, students, lecturers and the head of the department, also contributes to the assessment of these aspects.

4.5.4. Metodologias de ensino previstas com vista a facilitar a participação dos estudantes em atividades científicas (quando aplicável):

Os estudantes tomam contacto com regularidade com as atividades de investigação do centro de investigação associado ao Departamento (NOVA LINCS) e com a investigação em Informática. O Departamento organiza desde 2012 uma Distinguished Lecture anual, dada por um investigador de topo (incluimos vários Turing Awards) e à qual assistem todos os alunos do curso. Os docentes tendem a referir a sua investigação no contexto letivo. Uma das vertentes da UC Atividade de Desenvolvimento Curricular é um trabalho de introdução à investigação de 20 semanas, feito no seio de uma equipa de investigação, e apresentado numa workshop. A experiência demonstra que muitos dos estudantes que realizam esta vertente ficam ligados à investigação em que estiveram integrados, realizando uma tese de mestrado orientada para a investigação e, em alguns casos, o doutoramento.

4.5.4. Teaching methodologies that promote the participation of students in scientific activities (as applicable):

Students are in regular contact with the research activities of the Department's associated research center (NOVA LINCS) and with Computer Science research, in general. Since 2012, the Department has organized an annual Distinguished Lecture, given by a top researcher (including several Turing Awards) and attended by all students of the programme. Teaching staff tend to refer to their research in lectures. One of the aspects of the Curriculum Development Activity UC is a 20-week introductory research work done within NOVA LINCS and presented at a workshop. Experience shows that many of the students pursuing this option maintain a link to the research group in which they were integrated, carrying out a research-oriented master's thesis and, in some cases, entering the Department's PhD programme.

4.6. Fundamentação do número total de créditos ECTS do ciclo de estudos

4.6.1. Fundamentação do número total de créditos ECTS 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 DL n.º 74/2006, de 24 de março, com a redação do DL n.º 65/2018, de 16 de agosto:

De acordo com o Artigo 19.º do Decreto-Lei n.º 74/2006, e tratando-se de um 1º ciclo de estudos com 3 anos (6 semestres), foi atribuído ao ciclo de estudos um total de 180 ECTS para a obtenção do grau de Licenciado.

4.6.1. Justification of the total number of ECTS credits and of the duration of the study programme, based on articles 8 or 9 (1st cycle), 18 (2nd cycle), 19 (integrated master) and 31 (3rd cycle) of DL no. 74/2006, republished by DL no. 65/2018, of August 16th:

According to Article 8th and 9th of Decreto-Lei no. 74/2006, being the case of 1st cycle, we propose a 3-year duration (6 terms). This means that a total of 180 ECTS are needed for obtaining the BSc title.

4.6.2. Forma como os docentes foram consultados sobre a metodologia de cálculo do número de créditos ECTS das unidades curriculares:

Quase todas as UC do curso agora proposto existiam ou são atualizações de UC do atual Mestrado Integrado. O esforço do estudante nessas UC tem sido continuamente aferido e reajustado pelos docentes das mesmas sob a coordenação das Comissões Científicas, de forma informada por inquéritos na FCT NOVA, 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 foram auscultados sobre o método de cálculo das unidades ECTS.

4.6.2. Process used to consult the teaching staff about the methodology for calculating the number of ECTS credits of the curricular units:

Almost all of the UC in the proposed programme already existed (or are upgraded versions) of UC in the current Integrated Master. The students' efforts in these UC have been continuously measured and readjusted by their teaching staff under the coordination of the Scientific Committee of the programme, informed by surveys at FCT NOVA, since the beginning of the Bologna process. These periodic surveys enquire students about the number of hours they spend on the various activities and inform the process of adjusting ECTS credits. During the process of elaborating the forms of UC included in this proposal, the teaching staff were again actively involved and heard about the method of calculation of the ECTS credits of each UC.

4.7. Observações

4.7. Observações:

Pertencente ao Perfil Curricular FCT, está incluída nos planos curriculares, uma opção designada Unidade Curricular do Bloco Livre, a qual inclui unidades de todas as áreas científicas da FCT NOVA, com exceção das da área predominante do curso, aprovadas anualmente pelo Conselho Científico da FCT NOVA.

4.7. Observations:

As part of the FCT Curricular Profile, an option called Unrestricted Elective is included in the curricular plans, which includes curricular units from all scientific areas of FCT NOVA, with the exception of the predominant area of the course, approved annually by the Scientific Council of FCT NOVA.

5. Corpo Docente

5.1. Docente(s) responsável(eis) pela coordenação da implementação do ciclo de estudos.

5.1. Docente(s) responsável(eis) pela coordenação da implementação do ciclo de estudos.

João Alexandre Carvalho Pinheiro Leite / Nuno Manuel Ribeiro Pregoça

5.3 Equipa docente do ciclo de estudos (preenchimento automático)

5.3. Equipa docente do ciclo de estudos / Study programme's teaching staff

Nome / Name	Categoria / Category	Grau / Degree	Especialista / Specialist	Área científica / Scientific Area	Regime de tempo / Employment regime	Informação / Information
-------------	----------------------	---------------	---------------------------	-----------------------------------	-------------------------------------	--------------------------

Herberto de Jesus da Silva	Professor Auxiliar ou equivalente	Doutor	Matemática	100	Ficha submetida
Maria Cecília Perdigão Dias da Silva	Professor Auxiliar ou equivalente	Doutor	Matemática/ Álgebra	100	Ficha submetida
José Maria Nunes de Almeida Gonçalves Gomes	Professor Auxiliar ou equivalente	Doutor	Análise Matemática	100	Ficha submetida
António Maria Lobo César Alarcão Ravara	Professor Associado ou equivalente	Doutor	Matemática	100	Ficha submetida
Artur Miguel de Andrade Vieira Dias	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Maria Armada Simenta Rodrigues Grueau	Professor Auxiliar ou equivalente	Doutor	Information Science	100	Ficha submetida
Bernardo Parente Coutinho Fernandes Toninho	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Luís Filipe Santos Gomes	Professor Associado ou equivalente	Doutor	Engenharia Electrotécnica	100	Ficha submetida
Ruy Araújo da Costa	Professor Associado ou equivalente	Doutor	Engenharia de Sistemas	100	Ficha submetida
Miguel Jorge Tavares Pessoa Monteiro	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
João Jorge Ribeiro Soares Gonçalves de Araújo	Professor Catedrático ou equivalente	Doutor	Matemática	100	Ficha submetida
Ana Luísa da Graça Batista Custódio	Professor Associado ou equivalente	Doutor	Matemática	100	Ficha submetida
Jorge Manuel Leocádio André	Professor Auxiliar ou equivalente	Doutor	Matemática	100	Ficha submetida
Pedro Abílio Duarte de Medeiros	Professor Associado ou equivalente	Doutor	Informática	100	Ficha submetida
Hervé Miguel Cordeiro Paulino	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Vítor Manuel Alves Duarte	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Paulo Orlando Reis Afonso Lopes	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Carla Maria Gonçalves Ferreira	Professor Associado ou equivalente	Doutor	PhD in Computer Science	100	Ficha submetida
Maria Cecília Farias Lorga Gomes	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Miguel Carlos Pacheco Afonso Goulão	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
João Miguel da Costa Magalhães	Professor Associado ou equivalente	Doutor	Doutoramento em Computer Science	100	Ficha submetida
Fernanda Maria Barquinha Tavares Vieira Barbosa	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Rui Pedro da Silva Nóbrega	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
António Carlos Simões Paiva	Professor Auxiliar ou equivalente	Doutor	Física Atómica e Molecular	100	Ficha submetida
Pedro Manuel Corrêa Calvente Barahona	Professor Catedrático ou equivalente	Doutor	Informática (Computer Science)	100	Ficha submetida
Joaquim Francisco Ferreira da Silva	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Jorge Carlos Ferreira Rodrigues da Cruz	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Nuno Miguel Cavalheiro Marques	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
José Luís Toivola Câmara Leme	Professor Auxiliar ou equivalente	Doutor	Epistemologia das ciências	100	Ficha submetida
Carlos Augusto Isaac Piló Viegas Damásio	Professor Associado ou equivalente	Doutor	Informática	100	Ficha submetida
Jorg Matthias Knorr	Professor Auxiliar ou equivalente	Doutor	Doutoramento em Informática	100	Ficha submetida
Ricardo João Rodrigues Gonçalves	Professor Auxiliar ou equivalente	Doutor	Matemática	100	Ficha submetida
Fernando Pedro Reino da Silva Birra	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
João Ricardo Viegas da Costa Seco	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Maria de Fátima Varregoso Miguens	Professor Auxiliar ou equivalente	Doutor	Probabilidades e Estatística	100	Ficha submetida

Nuno Manuel Robalo Correia	Professor Catedrático ou equivalente	Doutor	Eng. Electr. e Comp.	100	Ficha submetida
Sofia Carmen Faria Maia Cavaco	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Vasco Miguel Moreira do Amaral	Professor Auxiliar ou equivalente	Doutor	Ciência da Computação	100	Ficha submetida
José Augusto Legatheaux Martins	Professor Catedrático ou equivalente	Doutor	Informática - Sistemas Distribuídos	100	Ficha submetida
João Alexandre Carvalho Pinheiro Leite	Professor Associado ou equivalente	Doutor	Informática	100	Ficha submetida
João Baptista da Silva Araújo Júnior	Professor Auxiliar ou equivalente	Doutor	Software Engineering	100	Ficha submetida
Margarida Paula Neves Mamede	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Henrique João Lopes Domingos	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Nuno Manuel Ribeiro Preguiça	Professor Associado ou equivalente	Doutor	Informática	100	Ficha submetida
João Carlos Antunes Leitão	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática e de Computadores	100	Ficha submetida
Sérgio Marco Duarte	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
				4600	

<sem resposta>

5.4. Dados quantitativos relativos à equipa docente do ciclo de estudos.

5.4.1. Total de docentes do ciclo de estudos (nº e ETI)

5.4.1.1. Número total de docentes.

46

5.4.1.2. Número total de ETI.

46

5.4.2. Corpo docente próprio - Docentes do ciclo de estudos em tempo integral

5.4.2. Corpo docente próprio – docentes do ciclo de estudos em tempo integral.* / "Full time teaching staff" – number of teaching staff with a full time link to the institution.*

Corpo docente próprio / Full time teaching staff	Nº / No.	Percentagem / Percentage
Nº de docentes do ciclo de estudos em tempo integral na instituição / No. of teaching staff with a full time link to the institution:	46	100

5.4.3. Corpo docente academicamente qualificado – docentes do ciclo de estudos com o grau de doutor

5.4.3. Corpo docente academicamente qualificado – docentes do ciclo de estudos com o grau de doutor* / "Academically qualified teaching staff" – staff holding a PhD*

Corpo docente academicamente qualificado / Academically qualified teaching staff	ETI / FTE	Percentagem / Percentage
Docentes do ciclo de estudos com o grau de doutor (ETI) / Teaching staff holding a PhD (FTE):	46	100

5.4.4. Corpo docente do ciclo de estudos especializado

5.4.4. Corpo docente do ciclo de estudos especializado / "Specialised teaching staff" of the study programme.

Corpo docente especializado / Specialized teaching staff

	ETI / FTE	Percentagem* / Percentage*	
Docentes do ciclo de estudos com o grau de doutor especializados nas áreas fundamentais do ciclo de estudos (ETI) / Teaching staff holding a PhD and specialised in the fundamental areas of the study programme	33	71.739130434783	46
Especialistas, não doutorados, de reconhecida experiência e competência profissional nas áreas fundamentais do ciclo de estudos (ETI) / Specialists not holding a PhD, with well recognised experience and professional capacity in the fundamental areas of the study programme	0	0	46

5.4.5. Estabilidade e dinâmica de formação do corpo docente.**5.4.5. Estabilidade e dinâmica de formação do corpo docente. / Stability and development dynamics of the teaching staff**

Estabilidade e dinâmica de formação / Stability and training dynamics	ETI / FTE	Percentagem* / Percentage*	
Docentes do ciclo de estudos em tempo integral com uma ligação à instituição por um período superior a três anos / Teaching staff of the study programme with a full time link to the institution for over 3 years	46	100	46
Docentes do ciclo de estudos inscritos em programas de doutoramento há mais de um ano (ETI) / FTE number of teaching staff registered in PhD programmes for over one year	0	0	46

Pergunta 5.5. e 5.6.**5.5. Procedimento de avaliação do desempenho do pessoal docente e medidas conducentes à sua permanente atualização e desenvolvimento profissional.**

O Regulamento da FCT NOVA relativo à Avaliação do Desempenho têm por objeto o desempenho dos docentes, visando avaliá-lo em função do mérito e melhorar a sua qualidade. A avaliação de desempenho abrange todos os docentes das escolas envolvidas, tem em conta a especificidade de cada área disciplinar e considera todas as vertentes da respetiva atividade: a) Docência; b) Investigação científica, desenvolvimento e inovação; c) Tarefas administrativas e de gestão académica; d) Extensão universitária, divulgação científica e prestação de serviços à comunidade. Os resultados da avaliação têm consequências no posicionamento remuneratório, contratação por tempo indeterminado e renovações de contratos. Para a permanente atualização dos docentes contribui, desde logo, a implementação de uma política de estímulo à investigação de qualidade com o objetivo de incentivar projetos com potencial de investigação e reconhecer o mérito dos investigadores mais destacados.

5.5. Procedures for the assessment of the teaching staff performance and measures for their permanent updating and professional development.

The Evaluation of the Performance's Statutes of FCT NOVA evaluate the merit of all academic staff, in order to improve their quality. The evaluation considers the specificities of each scientific area and aims at all the aspects of academic activity: a) Teaching; b) Research, development and innovation; c) Administrative work and academic management; d) Dissemination and community support activities. The evaluations' results impact the remuneration of the academic staff, tenure, contract renewal of professors, authorisation of sabbatical leaves, teaching load, and grants. The implementation of incentives for quality research based on the evaluation, contributes to continuous updates of staff, to improve the research potential, and to acknowledge the merit of the most recognised professors.

5.6. Observações:

<sem resposta>

5.6. Observations:

<no answer>

6. Pessoal Não Docente**6.1. Número e regime de tempo do pessoal não-docente afeto à leção do ciclo de estudos.**

*Participam em atividades de suporte à gestão do ciclos de estudos o seguinte pessoal não docente:
Gracinda Caetano – Coordenadora administrativa – Técnica Superior
Anabela Fonseca Nunes da Silva Duarte - Assistente Administrativa
Sandra Rosa Rafael Rainha - Assistente Administrativa
Apoio à coordenação e disseminação do curso, apoio aos docentes, atendimento dos estudantes.*

*Joana Dâmaso – Bolseira de Gestão de C&T
Ligação com o NOVA LINCS.*

*Bruno Machado - 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.*

6.1. Number and work regime of the non-academic staff allocated to the study programme.

The following non academic staff participate in management support activities related to the study cycle:

*Gracinda Caetano – Administrative Coordinator – “Técnica Superior”
Anabela Fonseca Nunes da Silva Duarte – Administrative Assistant
Sandra Rosa Rafael Rainha - Administrative Assistant*

*Joana Dâmaso – Bolseira de Gestão de C&T
Articulation with Research activities and NOVA LINCS.
Support to program coordination and dissemination activities, support to the teaching staff, assistance and administrative interface with the students.*

*Bruno Machado– 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.*

6.2. Qualificação do pessoal não docente de apoio à leção do ciclo de estudos.

*Gracinda Caetano – Licenciatura.
Anabela Fonseca Nunes da Silva Duarte – Ensino Secundário.
Sandra Rosa Rafael Rainha - Ensino Secundário
Joana Dâmaso – Mestrado
Bruno Machado – Ensino Secundário.*

6.2. Qualification of the non-academic staff supporting the study programme.

*Gracinda Caetano - BSc
Anabela Fonseca Nunes da Silva Duarte – Secondary Level
Sandra Rosa Rafael Rainha - Secondary Level
Joana Dâmaso – MSc
Bruno Machado – Secondary Level*

6.3. Procedimento de avaliação do pessoal não-docente e medidas conducentes à sua permanente atualização e desenvolvimento profissional.

A avaliação do pessoal não docente é efetuada segundo o SIADAP – Sistema Integrado de Avaliação de Desempenho da Função Pública – o qual assenta na definição de objetivos institucionais que são desdobrados pela organização. Os objetivos a atingir por cada funcionário, administrativo ou técnico, são definidos no início de cada ciclo avaliativo e estão alinhados com os objetivos estratégicos da instituição. A progressão do funcionário, a existir, dependerá da avaliação bienal que é feita em função do cumprimento das metas fixadas.

6.3. Assessment procedures of the non-academic staff and measures for its permanent updating and personal development

The performance of non-academic staff is based on SIADAP – Integrated System for Performance Evaluation of Public Administration. SIADAP requires the definition and deployment of institutional objectives. The goals to be attained by the non-academic staff are aligned with the institution strategic objectives and are defined at the beginning of each evaluation cycle. The career progression of staff depends on their biennial evaluation, which is based on the degree of accomplishment of the pre-defined goals.

7. Instalações e equipamentos

7.1. Instalações físicas afetas e/ou utilizadas pelo ciclo de estudos (espaços letivos, bibliotecas, laboratórios, salas de computadores, etc.):

*38 Gabinetes docentes e investigadores (área 830m2/lotação máxima 80 pessoas)
3 Salas de reuniões com datashow de acesso livre (área 64m2/27 lugares-9 por sala)
1 Sala de reuniões departamental com datashow e videoconferência (área 80 m2/20 lugares)
1 Sala de seminários com datashow e videoconferência (área 55m2/50 lugares)*

7 *Gabinets and rooms for administrative and technical staff (area 170 m²/occupation of 9 people)*
 6 *Server rooms and IT infrastructure (area total 100m²)*
 8 *Laboratories for research (area 300m²/maximum occupation 73 collaborators)*
 10 *Laboratories for Informatics teaching with datashow (area 700m²/300 places - 30 per room), including specific laboratories for networks and student projects.*
 6 *Open student rooms with computer or work table (area 170m²/88 places)*
 34 *Access control infrastructure based on magnetic cards.*
 10 *Video-surveillance cameras*
Various common spaces; Central Library FCT NOVA, Cafeterias, etc.

7.1. Facilities used by the study programme (lecturing spaces, libraries, laboratories, computer rooms, ...):

38 *Offices for teaching and research staff (area 830m²/capacity of 80 people)*
 3 *Free access meeting rooms with datashow (area 64m²/capacity 9 people / room)*
 1 *Departmental meeting room with datashow and videoconferencing (area 80 m²/capacity 20 people)*
 1 *Seminar room with datashow and videoconferencing (area 55m²/capacity 50 people)*
 7 *Offices and rooms for technical and administrative staff (area 170 m²/capacity 9 people)*
 6 *Server rooms and IT infrastructure space (area 100m²)*
 8 *Research Labs (area 300m²/capacity of 73 people)*
 10 *Teaching Labs for students work and classes with datashow (area 700m²/capacity 300 places - 30/room) including specialized laboratories for computer networks and student projects.*
 6 *Free access student rooms with computer or working table (total area 170m² total capacity 88 places)*
 34 *Access control infrastructure based on magnetic cards.*
 10 *Surveillance video-cameras.*
Several common spaces; Central Library FCT NOVA, Cafeterias, etc.

7.2. Principais equipamentos e materiais afetos e/ou utilizados pelo ciclo de estudos (equipamentos didáticos e científicos, materiais e TIC):

18 *Equipment data show (fixed)*
 3 *Equipment data show (portable)*
 190 *Computers (for students)*
 70 *Computers (teachers)*
 10 *Computers (technical and administrative services)*
 19 *Computers (central services)*
 12 *Computers aggregated in cluster (students)*
 110 *Computers aggregated in cluster (research)*
 17 *Network infrastructures without cables (APs)*
 34 *Network infrastructures with cables (switches)*
 2 *Firewalls*
 4 *Printers for teachers, services and post-graduate*
 2 *Photocopiers*
 2 *Video conferencing equipment; 1 Sound system*
 2 *Plasma TVs*
 15 *Tablet computers*
 60 *TFT 22" monitors (for laptops)*

7.2. Main equipment or materials used by the study programme (didactic and scientific equipment, 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 in cluster configuration (students)*
 110 *Servers 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)*

8. Atividades de investigação e desenvolvimento e/ou de formação avançada e desenvolvimento profissional de alto nível.

8.1. Centro(s) de investigação, na área do ciclo de estudos, em que os docentes desenvolvem a sua atividade científica

8.1. Mapa VI Centro(s) de investigação, na área do ciclo de estudos, em que os docentes desenvolvem a sua atividade científica / Research centre(s) in the area of the study programme where teaching staff develops its scientific activity

Centro de Investigação / Research Centre	Classificação (FCT) / Classification FCT	IES / HEI	N.º de docentes do CE integrados / Number of study programme teaching staff integrated	Observações / Observations
NOVA LINCS - NOVA Laboratory for Computer Science and Informatics	Excelente / Excellent	Faculdade de Ciências e Tecnologia da Universidade NOVA de Lisboa	31	http://nova-lincs.di.fct.unl.pt/

Pergunta 8.2. a 8.4.

8.2. Mapa-resumo de publicações científicas do corpo docente do ciclo de estudos, em revistas de circulação internacional com revisão por pares, livros ou capítulos de livro, relevantes para o ciclo de estudos, nos últimos 5 anos.

<http://www.a3es.pt/si/iportal.php/cv/scientific-publication/formId/9d92f453-bf36-1bca-344a-5e73c3187e5e>

8.3. Mapa-resumo de atividades de desenvolvimento de natureza profissional de alto nível (atividades de desenvolvimento tecnológico, prestação de serviços ou formação avançada) ou estudos artísticos, relevantes para o ciclo de estudos:

<http://www.a3es.pt/si/iportal.php/cv/high-level-activities/formId/9d92f453-bf36-1bca-344a-5e73c3187e5e>

8.4. Lista dos principais projetos e/ou parcerias nacionais e internacionais em que se integram as atividades científicas, tecnológicas, culturais e artísticas desenvolvidas na área do ciclo de estudos.

O DI e NOVA LINCS têm estado sempre envolvidos em redes e projetos nacionais (FCT/MCTES, QREN, ADI, CoLABs), da UE (ERC, H2020, Strep, IP, NoE, COST), e outros como os programas Carnegie-Mellon e UT Austin Portugal e o Mestrado Europeu em Lógica Computacional.

O DI/NOVA LINCS está envolvido em mais de 30 projetos financiados, alguns por empresas e participa em 4 CoLABs. Colaboramos regularmente com escolas nacionais (U. Lisboa, U. Minho, U. Porto, U. Aveiro, U. Coimbra, U. Évora, U. Madeira, U. Beira Interior e outras).

Alunos de doutoramento do DI têm sido coorientados com Carnegie-Mellon, UT Austin, IMT Lucca, TUKaiserlautern, Wright U., Columbia U., UFRGS. Decorrem colaborações com muitas universidades e empresas internacionais e nacionais (INRIA, Google, Imperial College, Dresden U, TUKaiserlautern, ESA, OutSystems, Altran, NOS, Edisoft, Farfetch, José de Mello Saúde, Vodafone Portugal, Fraunhofer Portugal, Centro Hospital de Lisboa Central, INIAV, Bayer CropScience, Syngenta Crop Protection, etc.). A lista completa de projetos pode ser consultada no site web do NOVA LINCS (<http://nova-lincs.di.fct.unl.pt>).

8.4. List of main projects and/or national and international partnerships underpinning the scientific, technologic, cultural and artistic activities developed in the area of the study programme.

The Department and NOVA LINCS have been involved in research networks and projects, national (FCT/MCT, QREN, ADI, CoLABs), EU (ERC, H2020, Strep, IP, NoE, COST), and others such as the Carnegie-Mellon and UT Austin partnerships, the European Master in Computational Logic.

DI/NOVA LINCS is currently involved in more than 30 funded projects, some by companies, and participates in 4 CoLABs. We collaborate regularly with researchers of other national schools (IST, U. Lisboa, U. Minho, U. Porto, U. Aveiro, U. Coimbra, U. Évora, U. Madeira, U. Beira Interior and others).

Several of our PhD students are co-supervised with institutions such as Carnegie Mellon, UT Austin, IMT Lucca, TU Kaiserslautern, Wright University, Columbia U. and UFRGS. Many academic and industrial collaborations are ongoing (INRIA, Google, Imperial College, Dresden U, TU Kaiserslautern, ESA, OutSystems, Altran, NOS, Edisoft, Farfetch, José de Mello Saúde, Vodafone Portugal, Fraunhofer Portugal, Centro Hospital de Lisboa Central, INIAV, Bayer CropScience, Syngenta Crop Protection, etc.). A complete list of projects and partnerships may be found in the web site of NOVA LINCS (<http://nova-lincs.di.fct.unl.pt>).

9. Enquadramento na rede de formação nacional da área (ensino superior público)

9.1. Avaliação da empregabilidade dos graduados por ciclo de estudos similares com base em dados oficiais:

Desde 1999, o DI FCT NOVA graduou mais de 1700 licenciados e mestres. Os dados disponíveis no Portal Infocursos da DGES só começam em 2015 e todos dizem respeito ao Mestrado Integrado (MIEI). Atualmente o MIEI tem cerca de 240 estudantes em cada um dos anos do 1.º ciclo e cerca 200 em cada um dos anos do 2.º ciclo, estimando-se em 80% o

número de estudantes que prosseguem para o mestrado. Para os diplomados do mestrado, de acordo com os dados disponíveis, entre 2012/13 e 2015/16, encontravam-se registados como desempregados no IEFP 2,4%, correspondente a uma taxa de empregabilidade de 97,6%. De acordo com os mesmos dados, a média nacional dos cursos da área é de 2,5% de diplomados, correspondente a uma taxa de empregabilidade de 97,5%. De acordo com os dados coligidos no relatório do OBIPNOVA (Inquéritos efetuados em 2016 aos Diplomados de 2014 (1 ano após)), a percentagem de diplomados em Engenharia Informática da FCT NOVA com emprego ao fim de um ano é de 100%.

9.1. Evaluation of the employability of graduates by similar study programmes, based on official data:

Since 1999, DI of FCT NOVA has graduated more than 1700 BSc and MSc in Computer Science and Engineering. The data available on the DGES Infocursos Portal only begins in 2015 and thus, all refer to the Integrated Master (MIEI). Currently MIEI has about 240 students in the 2nd and 3rd years and 200 students in the 4th and 5th years, and it is possible to estimate in 80% the number of students who continue for the master's degree. For Masters graduates, according to official data available for students graduating between 2012/13 and 2015/16, 2.4% of graduates were registered as unemployed at IEFP, corresponding to an employability rate of 97.6%. According to the same data, the national average of courses in the area is 2.5% of graduates, corresponding to an employability rate of 97.5%. According to data collected in the OBIPNOVA report (2016 Graduate Surveys (1 year after)), the percentage of FCT NOVA Computer Engineering graduates with employment after one year is 100%.

9.2. Avaliação da capacidade de atrair estudantes baseada nos dados de acesso (DGES):

O DI FCT NOVA demonstrou ao longo de décadas a capacidade sustentada de atrair candidatos em qualidade e quantidade para a sua oferta educativa em Engenharia Informática. Tanto a Licenciatura pré-bolonha, como a antiga LEI e o atual MIEI preenchem há muitos anos todas as vagas disponíveis (170 em 2019). De acordo com a DGES, em 2018/19 253 colocaram o MIEI em 1.º lugar na sua candidatura, o que aconteceu para 90 dos colocados; o índice de procura (número de candidatos em 1.º opção a dividir pelo n.º de vagas) está muito próximo de 2. Conjugando estes dados, representativos dos conseguidos nos últimos anos, com os altos níveis de empregabilidade dos cursos na área da Informática, acredita-se que a Licenciatura em Engenharia Informática, primeiro passo de uma formação em Engenharia Informática de conceção, adquirida com o Mestrado, apresenta um alto potencial de atração.

9.2. Evaluation of the capability to attract students based on access data (DGES):

Over the decades, DI FCT NOVA has demonstrated sustained ability to attract a large number of highly-qualified candidates to its educational offer in Computer Science and Engineering. Both the Pre-Bologna Degree, the old LEI and the current MIEI have filled all available positions for many years (170 in 2019). According to DGES, in 2019/20, 253 students placed MIEI in 1st place in their application, which happened to 90 of those that entered MIEI; The demand index (number of first-choice candidates divided by the number of vacancies is very close to 2. Combining these data - representative of those obtained in recent years - with the high levels of employability of Computer Science and Engineering (CSE) programmes, we believe that the proposed programme, the first step towards a conception-oriented master degree, has a high attraction potential.

9.3. Lista de eventuais parcerias com outras instituições da região que lecionam ciclos de estudos similares:

A nível da formação de 1.º ciclo não estão estabelecidas parcerias formais com IES da região. No entanto, o Departamento de Informática mantém colaborações estreitas com escolas congéneres na região como o Instituto Superior Técnico e 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. O Departamento de Informática mantém ainda colaborações estreitas com outras escolas do país (Universidade de Évora e Beira Interior) por via da participação de docentes destas escolas no centro de investigação NOVA LINCS.

9.3. List of eventual partnerships with other institutions in the region teaching similar study programmes:

At 1st cycle level, there are no formal partnerships with other teaching institutions in the region. However, DI FCT NOVA maintains close collaborations with similar schools in the region, such as the Instituto Superior Técnico and the Faculty of Sciences of the University of Lisbon. These links happen in the context of the evaluation of master dissertations, partnerships in research projects, thus promoting the maintenance of communication channels and experience exchange. DI FCT NOVA also maintains close collaborations with other schools in the country (University of Évora and Beira Interior) through the participation of teaching staff from these schools in the NOVA LINCS research center.

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 duração e estrutura semelhantes à proposta:

Em Portugal, a maior parte das escolas de engenharia oferece, na área de Informática, cursos de 3 anos. O curso proposto compara-se naturalmente com cursos europeus que concedem diplomas do 1.º ciclo e que têm uma duração de 3 anos; um exemplo é o Bachelor of Engineering do Imperial College de Londres e o Computer Science Bachelor da EPF

Lausanne, ambos com a duração de 3 anos. Há exceções como o do sistema universitário espanhol em que o BSc tem a duração de 4 anos. 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; é esta a abordagem da Licenciatura em Engenharia Informática do DI FCT NOVA que se assume como a 1.ª parte de uma formação em engenharia informática de conceção, a ser completada com o mestrado.

10.1. Examples of study programmes with similar duration and structure offered by reference institutions in the European Higher Education Area:

In Portugal, most engineering schools offer 3-year courses in the area of Computer Science and Engineering. The proposed course naturally compares with European courses which award 1st cycle diplomas and have a duration of 3 years; examples are the Bachelor of Engineering from Imperial College London and the Computer Science Bachelor from EPF Lausanne, both lasting 3 years. There are exceptions such as the Spanish university system where the BSc lasts for 4 years. Regardless of the formal organization of the courses, and without compromising the mobility recommended by Bologna, it seems to be consensual at European level that a conception engineer training is only completed in 4-5 years; This is the approach of the BSc programme in Computer Science and Engineering of DI FCT NOVA whose goal is to constitute the 1st part of a conception-oriented computer engineering degree, to be completed with the master's programme.

10.2. Comparação com objetivos de aprendizagem de ciclos de estudos análogos existentes em instituições de referência do Espaço Europeu de Ensino Superior:

Os três anos do curso proposto baseiam-se nas recomendações do ACM/IEEE CS Curriculum 2013. A formação do Engenheiro Informático segue uma organização geral típica em três níveis: básico (1-3 anos), de consolidação (3-4 anos) e especialização (4-5 anos). Em relação aos 2 primeiros níveis, existe bastante consenso sobre as matérias a incluir, que é semelhante ao de outros cursos do país e também ao dos cursos de referência europeus; outra característica comum, sobretudo em Portugal, é o pequeno número de UC opcionais. Um aspeto distintivo do curso é a UC “Atividade de Desenvolvimento Curricular” (ADC) do 6.º semestre, única a nível nacional em cursos de Engenharia Informática, uma mais valia no currículo dos nossos estudantes. Oferece aos estudantes uma experiência integradora (estágio, investigação ou projeto), e que não existe na maioria dos cursos europeus sendo exceção a UC “Industrial Placement” do Imperial College, de objetivos semelhantes à vertente estágio de ADC.

10.2. Comparison with the intended learning outcomes of similar study programmes offered by reference institutions in the European Higher Education Area:

The organization of the proposed programme is based on the recommendations of the ACM/IEEE CS Curriculum 2013. In the programmes analyzed, Computer Science and Engineering (CSE) MSc follow a typical general organization with 3 levels: basic (1-3 yr), consolidation (3-4 yr) and specialization (4-5 yr). With regard to the first 2 levels, there is a broad consensus on the subjects to be included, which is similar to other programmes in reference institutions. Another common feature, especially in Portugal, is the small number of optional UC. A distinctive feature of the programme is the 6th semester “Curriculum Development Activity” (ADC) course, which is unique at a national level in CSE courses, an added value in our students' curriculum. It offers students an integrative experience (internship, research or project), which does not exist in most European programmes, one exception being the Imperial College UC Industrial Placement, with objectives similar to the ADC internship option.

11. Estágios e/ou Formação em Serviço

11.1. e 11.2 Estágios e/ou Formação em Serviço

Mapa VII - Protocolos de Cooperação

Mapa VII - Entidades

11.1.1. Entidade onde os estudantes completam a sua formação:

Entidades

11.1.2. Protocolo (PDF, máx. 150kB):

[11.1.2._Atividade de Desenvolvimento Curricular e PIPP.pdf](#)

Mapa VII - Acordo de Acolhimento Estagio MIEI

11.1.1. Entidade onde os estudantes completam a sua formação:

Acordo de Acolhimento Estagio MIEI

11.1.2. Protocolo (PDF, máx. 150kB):[11.1.2._Acordo de Acolhimento Estagio MIEI.pdf](#)**Mapa VII - Protocolo enquadramento Estagio MIEI****11.1.1. Entidade onde os estudantes completam a sua formação:***Protocolo enquadramento Estagio MIEI***11.1.2. Protocolo (PDF, máx. 150kB):**

<sem resposta>

11.2. Plano de distribuição dos estudantes**11.2. Plano de distribuição dos estudantes pelos locais de estágio e/ou formação em serviço demonstrando a adequação dos recursos disponíveis.(PDF, máx. 100kB).**[11.2._Plano de distribuição dos estudantes.pdf](#)**11.3. Recursos próprios da Instituição para acompanhamento efetivo dos seus estudantes nos estágios e/ou formação em serviço.****11.3. Recursos próprios da Instituição para o acompanhamento efetivo dos seus estudantes nos estágios e/ou formação em serviço:***A unidade curricular de Atividade de Desenvolvimento Curricular vertente Estágio tem um docente atribuído que coordena todo o processo, sendo o responsável pelo lançamento de notas e de verificação de qualidade dos estágios realizados.**Cada estágio tem um orientador na empresa e um docente elemento de ligação no DI FCT NOVA. Este último funciona como ponto de ligação, mantendo reuniões regulares com alunos e empresa, e também como avaliador do estudante na Unidade Curricular.**O número de estágios está limitado a cerca de 60, permitindo que no máximo cada docente tenha 2 alunos sob sua alçada, garantindo um acompanhamento apropriado.***11.3. Institution's own resources to effectively follow its students during the in-service training periods:***The internships of the Curricular Development Activity unit have an assigned faculty member who coordinates the entire process, being responsible for recording the grades and quality verification of the internships carried out. Each internship has a supervisor in the company and a DI FCT NOVA faculty member who accompanies the whole process. The latter functions as a liaison point, maintaining regular meetings with students and the company, and also as a student evaluator in the curricular unit. The number of internships is limited to around 60, allowing a maximum of each faculty member to have 2 students under his / her supervision, ensuring appropriate monitoring.***11.4. Orientadores cooperantes****11.4.1. Mecanismos de avaliação e seleção dos orientadores cooperantes de estágio e/ou formação em serviço, negociados entre a instituição de ensino superior e as instituições de estágio e/ou formação em serviço (PDF, máx. 100kB).****11.4.1 Mecanismos de avaliação e seleção dos orientadores cooperantes de estágio e/ou formação em serviço, negociados entre a instituição de ensino superior e as instituições de estágio e/ou formação em serviço (PDF, máx. 100kB).**[11.4.1_Mecanismos de avaliação .pdf](#)**11.4.2. Orientadores cooperantes de estágio e/ou formação em serviço (obrigatório para ciclo de estudos com estágio obrigatório por lei)****11.4.2. Mapa X. Orientadores cooperantes de estágio e/ou formação em serviço (obrigatório para ciclo de estudos com estágio obrigatório por Lei) / External supervisors responsible for following the students' activities (mandatory for study programmes with in-service training mandatory by law)**

Nome / Name	Instituição ou estabelecimento a que pertence / Institution	Categoria Profissional / Professional Title	Habilitação Profissional (1)/ Professional qualifications (1)	Nº de anos de serviço / Nº of working years
----------------	--	--	--	--

<sem resposta>

12. Análise SWOT do ciclo de estudos

12.1. Pontos fortes:

Experiência de mais de 40 anos do DI FCT NOVA no ensino da Engenharia Informática. Elevada empregabilidade de engenheiros informáticos. Todos os docentes a tempo integral doutorados e membros de um centro de investigação com classificação de Excelente. 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 de estudantes do curso em atividades de investigação. Boa ligação com empresas / empregadores, facilitando o envolvimento de estudantes em “internships”. Excelente apreciação da formação do DI, quer pelos empregadores que contactam com os estudantes na UC Atividade de Desenvolvimento Curricular, quer pelos alunos (inquéritos). Cobertura de competências transversais e “Perfil Curricular FCT”.

12.1. Strengths:

More than 40 years of experience of DI FCT NOVA in the teaching of Computer Science and Engineering. High employability of BSc and MSc graduates. All full-time PhD faculty and members of a research center rated Excellent. Excellent teaching and research infrastructures. Integration into national and international research and advanced education networks. Quality of the Department's research, and the involvement of programme students in research activities. Good connections with companies / employers, facilitating the placement of students in internships. Excellent appreciation of the skills of BSc and MSc graduates by employers - who contact students at the UC Curricular Development Activity- as well as students (surveys). Soft skills coverage in “FCT Curricular Profile”.

12.2. Pontos fracos:

A escassez de recursos para atender ao reforço e renovação de recursos humanos. O reforço do número de docentes é crucial por duas razões: o elevado número de estudantes por docente (17) e a idade média superior a 50 anos do corpo docente. Apesar de nos últimos anos terem sido sempre recrutados para a formação em Engenharia Informática do DI FCT NOVA candidatos com nota de candidatura em geral superior a 15/20, 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 também um desequilíbrio de género que seria interessante atenuar.

12.2. Weaknesses:

Scarcity of resources to address the reinforcement and renewal of human resources; strengthening the number of teachers is crucial for two reasons: the high number of students per teacher (17) and the average age of over 50 years of faculty. Although in the last years, the programme candidates had always an overall application grade of over 15/20, a sustained effort to obtain more highly-qualified students, with better knowledge of the scientific area, must continue. There is also a gender imbalance that would be interesting to mitigate.

12.3. Oportunidades:

As TICs são uma das áreas em que o emprego mais vai crescer, estimando-se em 5 milhões o número de empregos criados nesta área entre 2015 e 2027. A orientação do programa para a formação integrada de engenheiros informáticos de conceção, facilitando o fluxo de alunos no sentido do 2.º ciclo, potencia a coesão e solidez da formação concedida ao longo dos 5 anos, sem prejuízo da mobilidade entre 1.º e 2.º 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 de pós-graduados. Melhor exposição dos alunos a atividades de realização prática e/ou interdisciplinar em ambiente externo à escola, permitida pela UC ADC e pelo “Perfil Curricular FCT” referido anteriormente.

12.3. Opportunities:

ICT is one of the areas where employment will grow the most, with an estimated 5 million jobs created in this area between 2015 and 2027. The programme's orientation towards integrated training of design computer engineers, facilitating the flow of students towards the 2nd cycle, enhances the cohesion and solidity of the training provided over the 3+2 years, without prejudice to the mobility between 1st and 2nd cycle assured by law. Updated curriculum based on ACM / IEEE 2013 and alignment with research, technology development, and current market. Early involvement of students in research activities, reinforcing the internal recruitment base of post-graduates. Better exposure of students to practical and / or interdisciplinary activities in an environment outside the school, allowed by the UC ADC and the adoption of the “FCT Curricular Profile” mentioned above.

12.4. Constrangimentos:

A diminuição do número de candidatos em geral e dos que têm notas de candidatura de nível excelente, que contribuem para catalisar o desempenho dos estudantes em geral. Existe alguma dificuldade, mesmo em algumas grandes escolas a nível internacional, em atrair mais estudantes de elevado potencial para a Engenharia Informática, em comparação quer com outras Engenharias quer com outras áreas como Gestão. 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, incluindo pelo DI FCT NOVA, existe em muitos casos uma perceção deficiente do que é verdadeiramente a Engenharia Informática pelos estudantes de Ciência e Tecnologia.

12.4. Threats:

The decrease of the number of candidates to the programme, both globally and of those with excellent application grades; these ones contribute to catalyzing average student's performance. There is some difficulty, even in some large international schools, in attracting more high potential students to Computer Science and Engineering (CSE) compared to other Engineering and other areas such as Management. The situation could benefit from a better visibility and knowledge of the area in secondary education, and in society in general; Despite the efforts made, including by DI FCT NOVA, there is in many cases a poor perception of what CSE is really about by students of science and technology.

12.5. Conclusões:

A presente proposta de curso de Licenciatura em Engenharia Informática consubstancia uma visão atualizada sobre a formação do Engenheiro Informático (nível E1 de qualificação da Ordem dos Engenheiros). O objetivo do curso é formar Engenheiros Informáticos de banda larga, com uma formação coerente e relevante em termos da mobilidade prevista na lei e com bases para aceder a um 2.º Ciclo, onde irão completar a sua formação como Engenheiros Informáticos de conceção. O currículo, matérias e competências cobertas encontram-se atualizadas com base nas mais recentes recomendações ACM/IEEE 2013 em alinhamento com o desenvolvimento tecnológico, e as necessidades do mercado atual. Foi realizada uma análise sistemática de cobertura do currículo ACM/IEEE 2013, verificando-se uma cobertura quase completa dos temas de nível 1 que é estendida à cobertura de temas de nível 2

O curso proposto oferece aos estudantes oportunidades de atividade no exterior da escola, no contexto de protocolos de enquadramento e parcerias estabelecidas com empresas e potencia o envolvimento dos estudantes em atividades de investigação, reforçando a base de recrutamento interna das equipas de investigação do centro de investigação NOVA LINCS.

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. A adequação ao "Perfil Curricular FCT", comum a todos os cursos da escola, favorece o desenvolvimento de várias competências transversais, na área da comunicação e ciência tecnologia e sociedade.

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 como um primeiro passo para a formação integrada de engenheiros informáticos de conceção, facilita o fluxo de estudantes no sentido do 2.º ciclo e potencia a solidez e eficácia da formação concedida ao longo de 5 anos.

A empregabilidade dos cursos de Engenharia Informática da FCT NOVA 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 no país e no estrangeiro.

12.5. Conclusions:

The present proposal for a degree in Informatics Engineering is based on an updated view on the education of the Computer Engineer (E1 level of qualification of the Order of Engineers). The aim of the course is to train broadband Computer Science and Engineering specialists, with a coherent and relevant training, allowing both the mobility provided by law and the access to a 2nd Cycle, where they will complete their training as conception-oriented Computer Engineers. The curriculum, subjects and skills covered are up to date - based on the latest ACM/IEEE 2013 recommendations – and aligned with technological development and current market needs. A systematic analysis of ACM/IEEE 2013 curriculum coverage was performed, with almost complete coverage of tier 1 themes being extended to coverage of some tier 2 themes.

The proposed course offers students opportunities for activity outside the school, in the context of established partnerships with companies and enhances student involvement in research activities, reinforcing the internal recruitment base of research teams at the research center NOVA LINCS.

These objectives are aligned with the school's educational, scientific and cultural project. It is a strategic bet of the school to continue to contribute decisively to the development of advanced education and scientific research and innovation in Computer Science and Engineering, as a pioneer school in the area in Portugal. The adaptation to the "FCT Curricular Profile", common to all school courses, favors the development of various transversal competences in the area of communication and science, technology and society.

The curriculum is designed to prepare professionals to successfully evolve in the area of information technology, that is intensely subject to change. The orientation of the program as a first step towards the integrated training of conception-oriented engineers facilitates the flow of students towards the 2nd cycle and enhances the solidity and effectiveness of the training provided over 5 years.

The employability of Computer Engineering courses at FCT NOVA has been around 99%. It is believed that the proposal presented here offers an effective, up-to-date and timely response to the sustained need for increasingly qualified

Computer Science and Engineering professionals and is comparable with related programmes in reference institutions both in the country and abroad.