

NCE/19/1901047 — 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:

Mestre

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:

120

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):

2 anos (4 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):

2 years (4 semesters)

1.9. Número máximo de admissões:

170

1.10. Condições específicas de ingresso.

Licenciados em Engenharia Informática ou curso afim.

1.10. Specific entry requirements.

BSc degree in Computer Science or similar.

1.11. Regime de funcionamento.

Diurno

1.11.1. Se outro, especifique:

<sem resposta>

1.11.1. If other, specify:

<no answer>

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 mais de 40 anos, no ensino da Engenharia Informática e consubstancia uma visão atualizada sobre a formação do Engenheiro Informático de conceção.

O ciclo de estudos (curso) proposto tem como objetivo a concessão de formação e maturação compatíveis com a prática profissional da Engenharia Informática de Conceção, alinhada com o estabelecido na escola, para as Engenharias, com os pressupostos da acreditação da Ordem dos Engenheiros para o nível de qualificação E2 e do EUR-ACE, e com programas afins em várias escolas de referência.

A estrutura do curso está organizada pressupondo dos candidatos uma formação de base compatível com as recomendações do ACM/IEEE CS Curriculum 2013 para um primeiro ciclo, e visa consolidar e aprofundar as competências, orientadas para a inovação, para a conceção de serviços e produtos, e para investigação científica. O bloco base (30 ECTS, 25% do curso), oferece 10 UC de Informática no primeiro ano do curso, que cobrem matérias tipicamente classificadas no nível 2 ACM/IEEE. Pretende-se com estas UC conceder uma formação muito sólida, de banda larga, em Engenharia Informática. De forma a permitir alguma flexibilidade, sem comprometer o requisito de abrangência, os alunos terão que obter aprovação em 5 de entre as 10 UC oferecidas. Existem ainda 12 ECTS (10% do curso) em UC obrigatórias, na área da matemática, empreendedorismo e aspetos socioprofissionais da informática. Ao bloco base segue-se o bloco de especialização (30 ECTS, 25% do curso), constituído por UC opcionais, que cobrem matérias do nível 3 (elective) ACM/IEEE, permitindo ao estudante aprofundar conhecimentos numa subárea da Engenharia Informática. Os tópicos cobertos no bloco de especialização, e, em certa medida, no bloco base, complementam temas clássicos com tendências mais atuais, de acordo com as recomendações ACM/IEEE 2013, assegurando o alinhamento do curso com as necessidades da sociedade, e com a investigação atual na área. O curso culmina com uma Dissertação de Mestrado a realizar em ambiente académico ou académico-empresarial, ou num Projeto de Engenharia a realizar em ambiente empresarial, precedidos em ambos os casos por uma Preparação de Dissertação/Projeto (12+30 ECTS, 35% do curso).

1.14. Observations:

FCT NOVA pioneered Computer Science and Engineering (CSE) higher education in Portugal. This proposal builds on the experience of DI/FCT NOVA, accumulated in the teaching of Computer Science and Engineering for over 40 years and offers an updated vision on the educational path of a modern professional engineer in the CSE field.

The proposed programme aims to provide the skills and maturation compatible with the professional practice of conception-oriented engineering, aligned with the established standards in the school, for MSc Engineering degrees, with the requirements for the accreditation procedures to the level E2 qualification of "Ordem dos Engenheiros" and EUR-ACE, and with many similar programs in reference schools.

The course structure is organized assuming that the candidates have a base education compatible with the recommendations of the ACM/IEEE CS Curriculum 2013 for a first cycle and aims to consolidate and deepen innovation-oriented skills for the design of services and products, and for scientific research.

The core block (30 ECTS, 25% of the program) offers 10 Computer Science courses in the first year, which typically cover subjects classified under ACM/IEEE Tier 2. These courses are intended to provide very solid advanced conceptual design skills in a broad variety of CSE topics. In order to allow some flexibility without compromising the broadness requirement, students will need to pass 5 out of the 10 courses offered. Additionally, there are mandatory courses (12 ECTS, 10% of the program) in the areas of mathematics, entrepreneurship and social-professional aspects of informatics.

The core block is followed by the specialization block (30 ECTS, 25% of the program), with optional courses covering subjects of ACM/IEEE elective topics allowing the student to deepen the knowledge in a sub-area of CSE. Topics covered in the specialization block, and to some extent in the core block, complement classical themes with modern trends, in line with the ACM/IEEE 2013 recommendations, ensuring that the program is aligned with the needs of

society, and with current scientific research directions in the area. The program concludes with a Scientific Dissertation to be conducted in an academic or academic-industrial environment, or an Engineering Project to be conducted in an industrial environment, preceded in both cases by a Dissertation/Project Preparation (12 + 30 ECTS, 35% of the course).

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 8_MEI.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_MEI.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_MEI.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:

O objetivo do Mestrado em Engenharia Informática é completar a formação de Engenheiros Informáticos de conceção habilitados a desenvolver atividades de projeto, liderança, e inovação, por vezes em contexto de investigação, e com bases para aceder a um 3.º Ciclo. Como disposto no DL65/2018 para o grau de mestre, os graduados pelo curso deverão:

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

3.1. The study programme's generic objectives:

*The objective of the Master programme in CSE is to complete the education of conception-level Computer Science Engineers prepared to develop project, leadership, and innovation activities, many times in an R&D context, and with the basis to pursue a 3rd cycle degree. As stipulated in DL65/2018 for the master degree, programme graduates should be: Knowledgeable of principles, techniques and analytical skills supporting design and conceptual engineering activities, as well as technological and scientific innovation
Able to apply knowledge in the solution of new problems, including in multidisciplinary contexts
Able to research and integrate knowledge, and critically evaluate and compare solutions
Demonstrate the ability to reflect on the ethical and social consequences of their activity, in a perspective of Science,*

Technology and Society, and to communicate results with clarity and technical rigour-Develop a collaborative attitude and teamwork competencies, able to support lifelong learning skills

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

*Compreender os fundamentos e o estado-da-arte em Engenharia Informática de conceção, utilizar bases matemáticas e técnicas, e métodos científicos e de engenharia no desenvolvimento de atividades profissionais;
Lidar com as várias facetas dos sistemas informáticos, aliando capacidades orientadas para o desenvolvimento técnico, às de interpretação do contexto social em que estes se inserem, num espírito de 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, a partir de bases de preparação técnica e científica que possam alicerçar uma orientação para o empreendedorismo.*

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

*Understand the foundations and the state-of-the-art in Computer Science and Engineering, use suitable mathematical and technical instruments, as well as engineering and scientific methods in the professional activities;
Dominate the several facets of informatic systems, adding to technical development skills the ability to interpret the social context in which such systems are embedded, promoting a knowledge improvement, lifelong-learning attitude;
Deal with the complexity with pro activity, demonstrating creativity;
Work in a team, know how to document objectives, solutions and results, both to specialists and non- specialists;
Address ethical-professional aspects, including social / environmental responsibilities;
Report, summarize, and argue technical themes, develop a critical judgment on proposed solutions.
Develop a professional activity in CSE, building on a strong technical and scientific basis that may also be leveraged in entrepreneurship-oriented activities.*

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 Mestrado em Engenharia Informática contribui para a formação de Engenheiros com o nível adequado ao desenvolvimento de projeto e de atividades de inovação. Dada a crescente digitalização de todas as áreas da sociedade e a sua dependência de sistemas informáticos, existe uma crescente e sustentada procura de recursos humanos cada vez mais qualificados, necessários para a criação de produtos e serviços inovadores e competitivos, geradores de vantagens económicas. 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 mais de 40 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, ciência tecnologia e sociedade, e empreendedorismo, potenciando a ligação da escola à sociedade em geral através da oferta de atividades colaborativas com o exterior, em particular com as empresas, por exemplo através da oferta de estágios-internships. No contexto do mestrado, o "Perfil Curricular FCT" foca-se no empreendedorismo e na ligação às empresas através da oferta de Dissertações em regime académico-empresarial e de Projetos de Engenharia.

Finalmente, o curso contribui para formar estudantes preparados para aceder a um 3.º ciclo (Doutoramento) em Ciência e Engenharia Informática, potenciando a investigação realizada no Departamento de Informática e no seu Centro de Investigação. Para este objetivo contribui a possibilidade dos estudantes desenvolverem a sua dissertação em ambiente de investigação científica (mais de 500 dissertações desde 2009), da preocupação constante em dar visibilidade à fronteira do conhecimento nas unidades curriculares do mestrado, em particular as de especialização, e as oportunidades de envolvimento em atividades de investigação ao longo do seu percurso académico, incluindo o suporte do programa de Bolsas do DI para o efeito.

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 area of Science and Engineering, whose 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 proposed MSc programme contributes to the education of engineers in CSE with a level of competence suitable for project development and innovation activities. Given the increasing digitization of all areas of society and their reliance on computer systems, there is a growing and sustained demand for increasingly qualified human resources needed to create innovative and competitive products and services that can generate economic advantage. FCT NOVA is strategically strongly committed to continue contributing to the development of advanced education and research in Computer Science and Engineering, as it has done for over 40 years, as a pioneer school in the area in our country. It is worth mentioning the alignment of the programme with the so-called "FCT Curricular Profile", which favours the development of several transversal competences, in the area of communication, science, technology and society, and entrepreneurship, leveraging the connections between the school and the society at large, in particular, with firms and companies, for instance by offering to students internships. In the context of this master's degree, the "FCT Curricular Profile" focuses on entrepreneurship and the connection to firms and companies by offering Master Dissertations in academic-industrial environment or an Engineering Project conducted in an industrial setting.

Finally, the programme also contributes to the education of strong candidates to a 3rd cycle (PhD program) in Computer Science, leveraging on the research pursued in the CS Department and its Research Unit. To this objective contributes the possibility of students to develop their dissertation in the context of a departmental research activity (we have delivered more than 500 dissertations since 2009), the constant concern to give visibility to the frontier of knowledge in the programme's courses, in particular those of specialization, and the opportunities to engage in research activities throughout their academic career, including support from the department's scholarship program.

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	12	90	
Matemática / Mathematics	M	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	
(5 Items)		24	96	

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	Observações / Observations (5)
Introdução à Investigação Operacional / Introduction to Operations Research	M	Semestre 1/Semester1	168	T:28; PL:28	6	
Opção A1 / Option A1	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Opção A2 / Option A2	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Opção A3 / Option A3	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Unidade Curricular do Bloco Livre B / Unrestricted Elective B	QAC	Semestre 1/Semester1	168	depende da UC escolhida / dependent of choice	6	Optativa / Optional
Empreendedorismo / Entrepreneurship	CC	Trimestre 2/Quarter2	80	TP: 45	3	
Opção A4 / Option A4	I	Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Opção A5 / Option A5	I	Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Opção B1 / Option B1	I	Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Opção B2 / Option B2	I	Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Aspetos Socio-Profissionais da Informática / Social-Professional Aspects of Informatics (11 Items)	CHS	Semestre 2/Semester2	84	T:28	3	

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)
Opção B3 / Option B3	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Opção B4 / Option B4	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Opção B5 / Option B5	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Preparação de Dissertação / Projeto em Engenharia Informática / Preparation of MSc Dissertation / Project of Computer Science and Engineering	I	Semestre 1/Semester1	336	OT: 20	12	
Opção Avançada de Formação / Advanced Training Option (5 Items)	I	Semestre 2/Semester2	840	OT: 40	30	Optativa / Optional

Mapa III - - 1.º Ano - Grupo de Opções A - Bloco Base / 1st Year - Option Group A**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 - Grupo de Opções A - Bloco Base / 1st Year - Option Group A***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)
Aprendizagem Automática / Machine Learning	I	Semestre 1/Semester1	168	T:28; PL:28	6	O aluno deve realizar 30 ECTS nestas ou noutras UC da mesma área aprovadas em Conselho Científico
Concorrência e Paralelismo / Concurrency and Parallelism	I	Semestre 1/Semester1	168	T:28; PL:28	6	Students must perform 30 ECTS in these or other UC same area approved by the Scientific Council
Construção e Verificação de Software / Software Construction and Verification	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Interação Pessoa-Máquina / Human-Computer Interaction	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Interpretação e Compilação de Linguagens / Interpretation e Compilation of Programming Languages	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Modelação de Sistemas e Processos / Systems and Process Modelling	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Representação de Conhecimento e Sistemas de Raciocínio / Knowledge Representation and Reasoning Systems	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Sistemas de Bases de Dados / Databases Systems	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Sistemas de Computação em Cloud / Cloud Computing Systems	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Segurança de Redes e Sistemas de Computadores / Network and Computer Systems Security	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional

(10 Items)

Mapa III - - 1.º e 2.º Ano - Grupo de Opções B - Bloco de Especialização / 1st and 2nd Year - Option B Group**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 e 2.º Ano - Grupo de Opções B - Bloco de Especialização / 1st and 2nd Year - Option B Group***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 Sistemas Distribuídos / Algorithms and Distributed Systems	I	Semestre 1/Semester1	168	T:28; PL:28	6	O aluno deve realizar 30 ECTS nestas ou noutras UC da mesma área aprovadas em Conselho Científico
Conceção e Implementação de Aplicações para a Internet / Internet Application Design and Implementation	I	Semestre 1/Semester1	168	T:28; PL:28	6	Students must perform 30 ECTS in these or other UC same area approved by the Scientific Council
Engenharia Orientada a Modelos / Model-Driven Engineering	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Modelação de Dados / Data Modelling	I	Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional

		1/Semester1				
Programação com Restrições / Constraint Programming		Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Programação Concorrente: Linguagens e Técnicas / Concurrent Programming: Languages and Techniques		Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Qualidade do Software / Software Quality		Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Qualquer UC do Bloco Base / Any Option A		Semestre 1/Semester1	168	T:28; PL:28	6	Optativa / Optional
Aprendizagem Profunda / Deep Learning		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Arquitetura e Protocolos de Redes de Computadores / Computer Networks Architecture and Protocols		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Computação de Alto Desempenho / High Performance Computing		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Computação Multimédia / Multimedia Computing		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Confiabilidade de Sistemas Distribuídos / Dependable Distributed Systems		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Desenho de Algoritmos para Problemas de Otimização / Design of Algorithms for Optimization Problems		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Jogos e Simulação / Games and Simulation		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Mineração e Pesquisa de Dados Web / Web Data Mining and Search		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Processamento de Streams / Stream Processing		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Prospecção e Análise de Dados / Data Analytics and Mining		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Requisitos e Arquitetura de Software / Software Requirements and Architecture		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Segurança de Software / Software Security		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Sistemas de Computação Móvel e Ubíqua / Mobile and Pervasive Computing		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Teoria de Jogos Computacional / Computational Game Theory		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Visualização e Análise de Dados / Visualization and Data Analytics		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional
Web Geográfica / GeoWeb		Semestre 2/Semester2	168	T:28; PL:28	6	Optativa / Optional

(24 Items)**Mapa III - - 2.º Ano - Opção Avançada de Formação / 2nd Year - Advanced Training Option****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 - Opção Avançada de Formação / 2nd Year - Advanced Training Option***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)
Dissertação em Engenharia Informática / Dissertation in Computer Science and Engineering		Semestre 2/Semester2	840	OT: 40	30	Optativa / Optional
Projeto em Engenharia Informática / Project in Computer Science and Engineering		Semestre 2/Semester2	840	OT: 40	30	Optativa / Optional

(2 Items)

4.4. Unidades Curriculares

Mapa IV - Introdução à Investigação Operacional

4.4.1.1. Designação da unidade curricular:

Introdução à Investigação Operacional

4.4.1.1. Title of curricular unit:

Introduction to Operations Research

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:28; 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):

Ruy Araújo da Costa - T:56; PL:196

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 unidade curricular o estudante terá desenvolvido competências que lhe permitam:

- formular e resolver problemas de Programação Linear (Inteira);*
- identificar e resolver problemas básicos de Teoria da Decisão;*
- formular e resolver problemas de Filas de Espera;*
- gerar números pseudo-aleatórios e aplicá-los no contexto da Simulação de Filas de Espera.*

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

After this course a student should be able to:

- formulate and solve (Integer) Linear Programming problems;*
- identify and solve basic Decision Making problems;*
- formulate and solve Queueing problems;*
- generate pseudo-random numbers and use them to Simulate Queueing Systems.*

4.4.5. Conteúdos programáticos:

1 - Programação Linear:

Formulação de Problemas. Algoritmo do Simplex Revisto. Programação Linear Inteira: Algoritmo Branch and Bound. Algoritmo Transportes.

2 - Teoria da Decisão:

Decisão em Incerteza e Risco. Árvores de Decisão.

3 - Filas de Espera:

Modelos básicos de Filas de Espera, com distribuições não exponenciais e com Disciplina Prioritária. Redes de Filas de Espera.

4 - Simulação:

Métodos de geração de Números Pseudo-Aleatórios. Aplicações às Filas de Espera.

4.4.5. Syllabus:

1 - Linear Programming:

Formulating problems. Revised Simplex Algorithm. Linear Integer Programming: Branch and Bound Algorithm. Transportation Problem.

2 - Decision Theory:

Decisions under risk and under uncertainty. Decision Trees.

3 - Queueing Theory:

Basic models, non-Exponential distributions models, models with Priorities. Queueing Networks.

4 - Simulation:

Pseudo-Random Numbers Generation Methods. Applications to Queueing Theory.

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

As primeiras 6 a 7 semanas do semestre são dedicadas ao estudo da Programação Linear (Inteira), cobrindo os objetivos enunciados.

Os objetivos enunciados relativos à Teoria da Decisão são cobertos em 2 semanas.

O estudo das Filas de Espera é feito em 4 semanas.

A aplicação da Simulação às Filas de Espera é feita em 2 a 3 semanas.

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

The first 6 to 7 weeks of the semester are used to study (Integer) Linear Programming, covering its learning outcomes.

The introduction to Decision Making is done in 2 weeks.

Queueing Theory is studied in 4 weeks.

Application of Simulation to Queueing problems is done in 2 to 3 weeks.

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas teórico-práticas servem para a apresentação dos conteúdos do Programa.

Nas aulas práticas faz-se a aplicação de conceitos teóricos com a resolução de exercícios.

A classificação final na unidade curricular é a soma de 90% das classificações obtidas nos 2 Testes com a classificação da componente de Participação. Há defesa de nota (trabalho complementar e/ou oral) para classificações superiores a 17.

4.4.7. Teaching methodologies (including students' assessment):

Lectures will be used to introduce students to the main topics.

Labs allow students to apply theoretical concepts, solving exercises.

Final grade is the sum of 90% of the 2 Tests grades with the Participation grade. An additional project and/or oral exam is required for grades above 17.

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

A componente teórica necessária para atingir os objetivos de aprendizagem é ministrada nas aulas teóricas. As aulas práticas asseguram o contexto adequado para a sedimentação da aprendizagem.

A UC é apoiada com uma página moodle com testes semanais, que apoiam a aprendizagem.

É assegurado um horário de atendimento semanal, para apoiar os alunos.

Os requisitos de acesso a cada um dos testes e a obtenção de Frequência visam assegurar que os alunos acompanham regularmente a matéria e, assim, maximizam a sua probabilidade de sucesso na UC.

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

In order to satisfy the learning outcomes, theoretical aspects of the topics are addressed in the Lectures. Laboratory sessions ensure the adequate context for full understanding of the topics studied in this course.

This course has a moodle webpage with weekly Tests, to allow students to assess their learning.

There is a weekly office hours schedule, to support students learning.

The requirements to access each Test, as well as requirements to complete assessment (Frequência), are supposed to pressure students into a regular contact with the course, thus maximizing their success in the course.

4.4.9. Bibliografia de consulta/existência obrigatória:

1. *Introduction to Operations Research (1990 - 5ªEd.), Hillier, Lieberman - Mc Graw Hill*
2. *Operations Research - An Introduction(1992 - 5ª Ed.) Taha - Prentice Hall*
3. *"Elementos de apoio às aulas de IIO", "Enunciados de Exercícios de IIO", Ruy A. Costa*
4. *Investigação Operacional (1996), Valadares Tavares et al - Mc Graw Hill.*

Mapa IV - Empreendedorismo

4.4.1.1. Designação da unidade curricular:

Empreendedorismo

4.4.1.1. Title of curricular unit:

Entrepreneurship

4.4.1.2. Sigla da área científica em que se insere:

CC

4.4.1.3. Duração:

Trimestral / Trimester

4.4.1.4. Horas de trabalho:

80

4.4.1.5. Horas de contacto:

TP:45

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):

António Carlos Bárbara Grilo - TP:45

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 curso pretende motivar os estudantes para o empreendedorismo e para a necessidade da inovação tecnológica. O programa cobre vários tópicos que são importantes para a adoção de uma cultura aberta aos riscos suscitados em processos de criação de novos produtos ou atividades que exigem características empreendedoras. No final desta unidade curricular, os estudantes deverão ter desenvolvido um espírito empreendedor, uma atitude de trabalho em equipa e estar aptos a:

- 1) Identificar ideias e oportunidades para empreenderem novos projetos;*
- 2) Conhecer os aspetos técnicos e organizacionais inerentes ao lançamento dos projetos empreendedores;*
- 3) Compreender os desafios de implementação dos projetos (ex: mercado, financiamento, gestão da equipa) e encontrar os meios para os ultrapassar;*
- 4) Expor a sua ideia e convencer os stakeholders.*

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

This course is intended to motivate students for entrepreneurship and the need for technological innovation. It covers a list of topics and tools that are important for new venture creation as well as for the development of creative initiatives within existing enterprises. Students are expected to develop an entrepreneurship culture, including the following skills:

- 1) To identify ideas and opportunities to launch new projects;*
- 2) To get knowledge on how to deal with technical and organizational issues required to launch entrepreneurial projects;*
- 3) To understand the project implementation challenges, namely venture capital and teamwork management, and find the right tools to implement it;*
- 4) To show and explain ideas and to convince stakeholders.*

4.4.5. Conteúdos programáticos:

O empreendedorismo como estratégia de desenvolvimento pessoal e organizacional. Processos de criação de ideias. A proteção da propriedade intelectual: patentes e formalismos técnicos. A gestão de um projeto de empreendedorismo: planeamento; comunicação e motivação; liderança e gestão de equipas Marketing e inovação para o desenvolvimento de novos produtos e negócios. O plano de negócios e o estudo técnico-financeiro. Financiamento e Sistemas de Incentivos: formalidades e formalismos. A gestão do crescimento e o intraempreendedorismo.

4.4.5. Syllabus:

Strategy for entrepreneurship. Ideation and processes for the creation of new ideas. Industrial property rights and protection: patents and technical formalities. Managing an entrepreneurial project: planning; communication and motivation; leadership and team work. Marketing and innovation for the development of new products and businesses. Business plan and entrepreneurial finance. System of Incentives for young entrepreneurs. Managing growth and intrapreneurship.

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

O conteúdo programático foi desenhado para incentivar o estudante ao empreendedorismo e à perceção e análise da envolvente em busca de oportunidades de negócio, de forma a que consiga aplicar os conhecimentos adquiridos:

- 1) na transformação de conhecimento científico em ideias de negócio;*
- 2) na criação, seleção e desenvolvimento de uma ideia para um novo produto ou serviço;*
- 3) na elaboração de um plano de negócio e de um plano de marketing;*
- 4) na exposição das suas ideias em curto tempo e em ambientes stressantes.*

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

The syllabus was designed to encourage the student for entrepreneurship and for the perception and analysis of new business opportunities; with this program, the student may apply the knowledge provided:

- 1) to transform scientific knowledge in business ideas;*
- 2) to create, select and develop an idea for a new product or service;*
- 3) to draw a business plan and a marketing plan;*
- 4) to better explain and present its ideas in a short time and stressed environments.*

4.4.7. Metodologias de ensino (avaliação incluída):

Esta UC será ministrada a estudantes dos 1.º ou 2.º anos de 2.º ciclo. O programa é dimensionado para decorrer entre o 1.º e o 2.º semestre, num período de 5 semanas, envolvendo um total de 45 horas presenciais, organizadas em 15 sessões de 3 horas e exigindo um esforço global de 3 ECTS.

As aulas presenciais baseiam-se na exposição dos conteúdos do programa. Os estudantes serão solicitados a aplicar as competências adquiridas através da criação e desenvolvimento de uma ideia (produto ou negócio). As aulas integrarão estudantes provenientes de diversos cursos com vista a promover a integração de conhecimento derivado de várias áreas científicas e envolverão professores e "mentores" com background diverso em engenharia, ciência, gestão e negócios.

A avaliação compreende a apresentação e defesa da ideia num elevator pitch e do respetivo relatório (realizado em grupo de 4-5 elementos). A apresentação contribuirá com 60% e o relatório com 40% para a nota final.

4.4.7. Teaching methodologies (including students' assessment):

This course is directed to students from the 1th or 2nd years of the 2nd cycle (Master). The program was designed for a duration of 5 weeks, with a total of 45 hours in class (15 sessions of 3 hours each) - 3 ECTS.

Classes are based in an exposition methodology. Students will be asked to apply their skills in the creation and development of an idea, regarding a new product or a new business. Classes integrate students from different study programs to promote the integration of knowledge derived from various scientific areas and involve academic staff and "mentors" with diverse background in engineering, science, management and business.

Students evaluation is based on the development and presentation of an idea/project in an elevator pitch, and its report. The work should be developed in teams of 4-5 members. The presentation should account for 60% of the final mark and the report 40%.

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

Considerando o tempo disponível (5 semanas), a metodologia de ensino preconiza que em cada semana sejam discutidos e trabalhados (em grupo) os temas apresentados, os quais tinham sido definidos nos objetivos de aprendizagem.

Na 1.ª semana os temas a abordar estão relacionados com os aspetos estratégicos do empreendedorismo, a geração de ideias, a liderança e a gestão de equipas; como resultado os estudantes deverão constituir e organizar as suas equipas para poderem definir o problema que se pretende resolver. Na 2.ª semana, os temas apresentados permitirão que o estudante possa evoluir no seu projeto acrescentando opções de soluções ao problema identificado na semana interior e proceder à seleção de uma delas. Na 3.ª semana, a abordagem ao mercado e às condições de comercialização viabilizarão a concretização do plano de marketing.

Na 4.ª semana, abordar-se-ão os aspetos relacionados com a viabilidade financeira do projeto, possibilitando a realização do respetivo plano de negócio e do seu financiamento. Na última semana, abordar-se-á o processo de exposição da ideia aos potenciais interessados, tendo os estudantes que realizar a apresentação e defesa do seu projeto num elevator pitch, perante um júri.

Neste sentido, a metodologia privilegia:

- 1) a apresentação de casos práticos e de sucesso;
- 2) a promoção de competências nos domínios comportamentais, nomeadamente, no que respeita ao desenvolvimento do sentido crítico, à defesa de ideias e argumentos baseados em dados técnico-científicos, à tolerância e capacidade de gestão de conflitos em situações adversas e stressantes.
- 3) a participação dos estudantes nos trabalhos colocados ao longo da unidade curricular e a sua apresentação.

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

Considering the available time (5 weeks), the teaching methodology praises that, in each week, the subjects presented and defined in the learning objectives are discussed and worked (in groups). In the first week, the subjects introduced to students are related with entrepreneurial strategic issues, generation of ideas, leadership and work team management; as a result, the students will have to organize their teams to be able to define the problem. In the 2nd week, the subjects presented will allow the student to pursue its project; they have to consider different options for the problem identified in the previous week. In the 3rd week, the market related issues are approached, and the students are asked to build a marketing plan. In the 4th week, financial issues are addressed, making it possible to accomplish a business plan. In the last week, the process of how to expose the idea to potential stakeholders is addressed; the students are required to present and argue their project in an elevator pitch.

This methodology gives priority to:

- 1) the presentation of practical and successful cases;
- 2) the promotion of soft skills, namely, in what concerns to the development of critical thinking, the defense of ideas and arguments based on technical-scientific data, to the tolerance and capacity of dealing with conflicts in adverse and stressful situations.
- 3) the participation of the students in practical works and assessments and their presentation.

4.4.9. Bibliografia de consulta/existência obrigatória:

Books

Burns, P. (2010). Entrepreneurship and Small Business: Start-up, Growth and Maturity, Palgrave Macmillan, 3rd Ed.
Kotler, P. (2011). Marketing Management, Prentice-Hall
Shriberg, A. & Shriberg (2010). Practicing Leadership: Principles and Applications, John Wiley & Sons, 4th Ed.
Spinelli, S. & Rob Adams (2012). New Venture Creation: Entrepreneurship for the 21st Century. McGraw-Hill, 9th Ed.
Byers, Thomas H., Dorf R. C., Nelson, A. (2010). Technology Ventures: From Idea to Enterprise, 3rd Ed., McGraw-Hill
Hisrich, R. D. (2009). International Entrepreneurship: Starting, Developing, and Managing a Global Venture, Sage Publications, Inc
Hisrich, R.D., Peters, M. P., Shepherd, D.A. Entrepreneurship, 7th Ed., McGraw-Hill, 2007
Journals
Entrepreneurship Theory and Practice

Mapa IV - Aspetos Socio-Profissionais da Informática

4.4.1.1. Designação da unidade curricular:

Aspetos Socio-Profissionais da Informática

4.4.1.1. Title of curricular unit:

Social-Professional Aspects of Informatics

4.4.1.2. Sigla da área científica em que se insere:

CHS

4.4.1.3. Duração:

Semestral / Semester

4.4.1.4. Horas de trabalho:

84

4.4.1.5. Horas de contacto:

T:28

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 Augusto Legatheaux Martins - T:14***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Paulo Orlando Reis Afonso Lopes - T:14***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Saber:*

- *Impacto das comunicações e processamento digitais na sociedade;*
- *Ética e ética aplicada relacionadas com a utilização de sistemas modernos de comunicações e processamento digitais.*

Saber fazer:

- *Análise das controvérsias e impactos éticos do uso de comunicações e processamento digitais;*
- *Consciência das implicações da tecnologia na sociedade.*

Competência Complementares:

- *aprender a investigar temas das ciências sociais;*
- *capacidades de apresentação oral e escrita.*

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

- *Impact of communications and information technologies in the society;*
- *Ethics and applied ethics related to the use of modern communication and information systems.*

Application:

- *Analysis of controversies and ethical impacts of communication and information technology;*
- *Awareness on the social implications of technology.*

Soft Skills:

- *learn how to research social sciences subjects;*
- *oral and written presentation skills.*

4.4.5. Conteúdos programáticos:*Introdução - panorâmica da evolução da tecnologia digital e do impacto social da tecnologia**Moral, ética e lei**O impacto das comunicações digitais na privacidade e na interação humana**Segurança privacidade, hacking e cibercrime**Propriedade intelectual e plágio**Privacidade e proteção de dados individuais**O impacto da tecnologia digital na sociedade — focos: Internet e Inteligência Artificial**Redes sociais e o seu impacto social**Ética profissional***4.4.5. Syllabus:***Introduction - an overview of the evolution of digital technology and of the social impact of technology**Morals, ethics and the law**The impact of digital communications on privacy and human interaction**Security and privacy, computer hacking, cybercrime**Intellectual property and plagiarism**Privacy and protection of individual data**The impact of the digital in the society – highlights: Internet and AI**Social networks and their social impact**Professional Ethics***4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:***No início os estudantes são confrontados com a evolução exponencial recente das tecnologias digitais e do seu impacto social. Em seguida é apresentada uma introdução ao que é a moral, a ética e a lei, assim como são apresentadas as principais correntes éticas aplicáveis neste contexto.**Estes princípios éticos são depois usados para analisar alguns dos impactos emergentes das tecnologias digitais: privacidade, comunicações pessoais, segurança, propriedade intelectual, proteção de dados, inteligência artificial, redes sociais, ética profissional, etc.**Finalmente, os estudantes têm de realizar um trabalho de grupo sobre um dos temas propostos pelos professores e relacionados com o programa. A realização deste trabalho requer a realização de pesquisa de informação, escrever um artigo e apresentá-lo. Para apoiar esta fase, são igualmente realizados seminários sobre pesquisa de informação, citação e como evitar plágio.***4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:**

To start with, students are confronted with the recent exponential evolution of digital technologies and their social impact. Then follows an introduction to what is moral, ethics and the law, as well as a brief discussion of the main ethical theories relevant in this context.

These ethical principles are then used to analyze the most emergent impacts of digital technologies: privacy, personal communications, security, intellectual property, data protection, artificial intelligence, social networks, professional ethics, et cetera.

Finally, in addition to some tests or exam assessment, students also have to carry out a group work on one of the themes proposed by instructors and related to the syllabus. The accomplishment of this group work encompasses research of information, writing an article and its presentation. To this end, seminars on how to research information and the care one should have to avoid plagiarism are also presented.

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas da unidade curricular podem tomar as seguintes formas: aulas teóricas, seminários realizados por docentes ou individualidades convidadas, e visionamento de filmes (seguidos de debate). São ainda indicados artigos e temas para pesquisa que suportem o trabalho escrito.

A avaliação é realizada através de 2 testes teóricos e um trabalho final de grupo. A nota final é a média aritmética das 3 componentes.

4.4.7. Teaching methodologies (including students' assessment):

The course comprises any or all of the following: lectures, seminars conducted by instructors or invited speakers, and films (followed by debates). References to papers and themes for research will be provided to support the written group work.

Grading is based on two intermediate tests as well as a final home work. Final grading is the average of the 3 grading components.

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

As metodologias de ensino seguidas na unidade curricular são relativamente tradicionais neste tipo de assuntos.

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

The working methodologies of this course are quite common in this type of subjects.

4.4.9. Bibliografia de consulta/existência obrigatória:

Michael J. Quinn, "Ethics for the Information Age," Pearson, 7th Edition, 2010

Each theme will be covered with more references; the appropriate place to find them is the online documentation of the course.

Mapa IV - Preparação de Dissertação / Projeto em Engenharia Informática

4.4.1.1. Designação da unidade curricular:

Preparação de Dissertação / Projeto em Engenharia Informática

4.4.1.1. Title of curricular unit:

Preparation of MSc Dissertation / Project of Computer Science and 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:

336

4.4.1.5. Horas de contacto:

OT:20

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):***Carlos Augusto Isaac Piló Viegas Damásio - OT:40***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Todos os docentes do curso de Mestrado em Engenharia Informática - OT:40***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***O trabalho de dissertação de Mestrado / Projeto de Engenharia enquadra-se nos termos definidos na alínea b) do número 1 do Artigo 20.º do Decreto Lei n.º 65/2018, e consiste num trabalho individual de investigação e/ou desenvolvimento que explora os conhecimentos adquiridos ao longo do curso, original e especialmente concebido para esse fim. O objetivo desta UC consiste na realização de trabalho preparatório para a realização da dissertação / projeto de Engenharia Informática, incluindo o estudo do estado da arte e a elaboração dum plano de trabalho.***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***The dissertation work for the Master in Computer Science and Engineering conforms with the requirements of item b) of number 1, Article 20º of DL 65/2018, and consists of an individual research and / or development work which explores the knowledge acquired in the whole program, original, and specifically identified for that purpose. The goal of this CU consists in the development of preliminary work for the development of the dissertation / Project in Computer Science and Engineering, including a study of the state of the art and the elaboration of a work plan.***4.4.5. Conteúdos programáticos:***Desenvolvimento de trabalho conducente a elaboração de dissertação de mestrado em Engenharia Informática.***4.4.5. Syllabus:***Development of the work leading to the elaboration of a master dissertation in Computer Science and Engineering.***4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:***Os conteúdos são naturalmente coerentes com os objetivos, dada a sua natureza especial da unidade curricular.***4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:***Given the special nature of the curricular unit, the syllabus are naturally coherent with the unit's objectives.***4.4.7. Metodologias de ensino (avaliação incluída):***O trabalho conducente à dissertação de mestrado / projeto de Engenharia Informática desenrola-se em duas fases. Nesta UC desenrola-se a primeira fase (com a duração de 1 semestre, a tempo parcial de aproximadamente 40%), em que o estudante desenvolve o trabalho introdutório do tema da sua dissertação e apresenta um relatório incluindo um plano do trabalho a realizar para concluir a Dissertação de Mestrado / Projeto de Engenharia Informática, durante a segunda fase, a realizar na UC de Dissertação de Mestrado ou Projeto em Engenharia Informática. Esse relatório é discutido e avaliado numa sessão pública por um júri intermédio que o deverá validar e eventualmente apresentar sugestões relevantes.***4.4.7. Teaching methodologies (including students' assessment):***The work leading to the master dissertation / project in Computer Science Engineering develops in two consecutive phases. In this CU (with the duration of one term, but in partial time of approximately 40%) the student develops preliminary work for the dissertation / project theme and presents a written report, including a plan of the work needed to conclude the master dissertation / project work. This report is discussed and evaluated in a public session by an interim jury that validates and possibly proposes suggestions for improving the future work. Follows the second phase (duration of one term, with full time dedication).***4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular:***Tendo em conta a natureza especial desta unidade curricular, decorre que existe coerência adequada entre as metodologias de ensino e os objetivos de aprendizagem da mesma. Acresce que a metodologia de ensino adotada para a realização da dissertação / projeto de Engenharia Informática, estruturada em duas fases, com um ponto de aferição intermédio no fim desta UC, e que resulta da experiência de vários anos do Departamento de Informática na orientação de dissertações / projetos de mestrado, tem-se revelado um fator promotor do aumento de qualidade do processo de desenvolvimento das dissertações/projetos, na opinião de todos os envolvidos (estudantes, docentes, membros de júri, e parceiros exteriores). O trabalho de preparação, independentemente da sua natureza de introdução à investigação ou de projeto avançado de engenharia de conceção, poderá desenvolver-se quer em contexto académico, quer em contexto de colaboração académico-empresarial, em qualquer dos casos, sob coordenação científica geral da comissão científica de curso e orientação formal de um membro do corpo docente do curso. A avaliação da preparação incidirá sobre a prova oral e sobre o relatório, incluindo neste caso a avaliação da Organização e estrutura, estilo e correção da escrita,*

enquadramento do tema, trabalho relacionado, análise crítica do trabalho relacionado, qualidade do conteúdo técnico-científico, descrição da metodologia de avaliação dos resultados a obter, e plano de trabalhos.

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

Having in mind the special nature of this curricular unit, it clearly results that there is adequate coherence between the teaching methodologies and the learning outcomes. Additionally, we notice that the teaching methodology for the development of the dissertation / project in Computer Science and Engineering, structured in two phases, with an interim assessment point in the end of this CU, which results from the several years of experience of the Department on supervising master dissertations, has revealed itself as a factor that promotes a quality increase of the dissertation / project supervision process, in the opinion of all the stakeholders (students, faculty members, jury members, and external collaborators).

The preparation work, independently of its orientation towards research or advanced engineering project, may be developed either in academic context or in some collaboration between the department and an external organization (such as a firm), in any case under the scientific coordination of the program's scientific committee and formal supervision of a faculty member. The evaluation of this CU will focus on the oral presentation and discussion and the report, including in this case the valuation of the organization and structure, style and correction of the writing, context of the theme, related work, critical assessment of the related work, quality of the technical/scientific contents, description of the evaluation methodology, and work plan.

4.4.9. Bibliografia de consulta/existência obrigatória:

Depende de cada trabalho de dissertação/projeto de Engenharia / Depends on each specific dissertation/project work.

Mapa IV - Aprendizagem Automática

4.4.1.1. Designação da unidade curricular:

Aprendizagem Automática

4.4.1.1. Title of curricular unit:

Machine Learning

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Ludwig Krippahl – T:28; PL:84;

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):

Saber

- Compreender os paradigmas e desafios da área de Aprendizagem Automática. Aprendizagem Supervisionada, Aprendizagem Não-Supervisionada e Aprendizagem por Reforço.*
- Aprender métodos fundamentais e suas aplicações na descoberta de conhecimento orientada aos dados. Dados, seleção de modelos, complexidade de modelos, etc.*

- *Compreender vantagens e limitações dos métodos de Aprendizagem Automática estudados.*

Fazer

- *Implementar e adaptar algoritmos de Aprendizagem Automática.*
- *Modelar experimentalmente dados reais*
- *Interpretar e avaliar resultados experimentais.*
- *Validar e comparar algoritmos de Aprendizagem Automática.*

Competências Complementares

- *Capacidade de avaliar a adequação dos métodos a dados e aplicações práticas.*
- *Capacidade de avaliar criticamente os resultados obtidos.*
- *Autonomia para aplicar e aprofundar os conhecimentos na área de Aprendizagem Automática.*

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

Knowledge

- *Understand the paradigms and challenges of Machine Learning, distinguishing Supervised, Unsupervised and Reinforcement learning.*
- *Learn the fundamental methods and their applications in data oriented knowledge discovery. Understand data features, the selection of models and their complexity.*
- *Understand the advantages and disadvantages of the different methods.*

Applications

- *Implement and adapt Machine Learning algorithms;*
- *Model real data experimentally.*
- *Interpret and evaluate experimental results.*
- *Validate and compare different Machine Learning algorithms.*

Soft Skills

- *Evaluate the suitability of each method to concrete applications and data sets.*
- *Critical evaluation of the results.*
- *Autonomy and self-reliance in the application and furthering studies in Machine Learning.*

4.4.5. Conteúdos programáticos:

Introdução à Aprendizagem Automática.

Paradigmas de Aprendizagem Automática: Aprendizagem Supervisionada, Aprendizagem Não-Supervisionada e Aprendizagem por Reforço.

Dados

- 2.1 *Tipos de dados.*
- 2.2 *Medidas de proximidade e medidas de dispersão de dados.*
- 2.3 *Tópicos de normalização e visualização de dados*
- 2.4 *Visualização de Dados por Análise de Componentes Principais*

Aprendizagem Supervisionada

- 3.1 *Regressão*
- 3.2 *Árvores de Decisão*
- 3.3 *Redes Neurais*
- 3.4 *Máquinas de Suporte Vectorial*
- 3.5 *Modelos gráficos*
- 3.6 *Classificador dos K-Vizinhos mais Próximos*
- 3.7 *Avaliação e comparação de métodos de classificação*
- 3.8 *Ensembles*

Aprendizagem Não-Supervisionada

- 4.1 *Métodos de Agrupamento por partição*
- 4.2 *Métodos de Agrupamento Probabilístico*
- 4.3 *Métodos de Agrupamento Difuso por partição*
- 4.4 *Métodos de Agrupamento Hierárquico*
- 4.5 *Cadeias de Markov*
- 4.5 *Avaliação de métodos e de resultados de agrupamento*
- 4.6 *Outros métodos*

4.4.5. Syllabus:

Introduction to Machine Learning

Machine Learning paradigms: Supervised Learning, Unsupervised Learning and Reinforcement Learning.

Data

- 2.1 *Types of Data*
- 2.2 *Measures of similarity and dissimilarity*
- 2.3 *Data normalization and visualization*
- 2.4 *Dimensionality reduction by Principal Component Analysis*

Supervised Learning

- 3.1 *Regression*
- 3.2 *Decision Trees*
- 3.3 *Artificial Neural Networks*
- 3.4 *Support Vector Machines*
- 3.5 *Graphical models*
- 3.6 *K-nearest neighbour classifier*
- 3.7 *Methods for classifier evaluation and comparison*
- 3.8 *Ensembles*

Unsupervised Learning**4.1 Partitional clustering****4.2 Probabilistic clustering****4.3 Partitional Fuzzy clustering****4.4 Hierarchical clustering****4.5 Markov chain****4.6 Clustering evaluation methods****4.6 Other unsupervised learning topics****4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:**

O programa da disciplina, tanto na componente de Aprendizagem Supervisionada como na componente de Aprendizagem Não-Supervisionada, cobre todos os tópicos base e uma parte substancial dos tópicos considerados opcionais na sub-área 'Advanced Machine Learning' [elective] do Computer Science Curriculum 2013 da ACM (<http://cs2013.org/>).

As várias metodologias estudadas são enquadradas com exemplos de aplicação.

Os trabalhos práticos e exercícios tutoriais permitem consolidar conhecimentos na implementação/adaptação dos algoritmos estudados a problemas reais. Será dada particular atenção à interpretação de resultados experimentais, avaliação dos métodos usados e comparação de modelos.

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

Both the Supervised and Unsupervised Learning components of the syllabus cover all the core subjects and most of the elective subjects for Machine Learning in the ACM Computer Science Curricula 2013 (<http://cs2013.org/>).

The methods studied are applied to practical examples. Tutorial classes and practical assignments promote the consolidation of theoretical knowledge as well as practice in the implementation and adaptation of the algorithms to real life problems. Practical assignments also focus on the interpretation of experimental results and the evaluation of the methods used, including the comparison of different models.

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas cobrem os tópicos fundamentais da matéria, dedicando tempo para exposição da matéria e para dúvidas e discussão com os alunos.

As aulas laboratoriais destinam-se à orientação tutorial e realização dos trabalhos práticos.

Serão disponibilizados os acetatos da matéria teórica e uma página Web onde se mantém informação atualizada sobre o funcionamento da mesma.

A avaliação da disciplina é composta por uma componente teórica-prática (T) e uma componente laboratorial ou de projeto (P), tendo cada componente um peso de 50% na nota final. Para obter aprovação é necessário nota mínima de 10 valores em cada uma das componentes de avaliação e a nota final é a média das duas componentes de avaliação.

A componente teórica-prática (T) consiste em dois testes que contarão em partes iguais para a nota desta componente. Em alternativa, esta componente pode ser realizada por exame de recurso.

4.4.7. Teaching methodologies (including students' assessment):

Lectures will cover the fundamental topics of the subject matter and include some time for questions and discussion.

All lecture materials will be supplied for further study.

Tutorial classes will be dedicated to exercises and guidance in the practical assignments, focusing on selected topics.

Class schedules and materials will be supplied online, as well as additional information regarding the course.

Assessment

The evaluation of this curricular unit is made by two components: theoretical/problems (T) and project (P). Each component contributes with 50% to the final grade.

To pass, the student must have: a grade of at least 10 points in each of the theoretical/problems and project components.

The final grade is defined as the weighted average of the two components of evaluation.

The theoretical part consists of two written individual tests; alternatively, this component can be evaluated by a written exam.

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

Os objetivos desta unidade curricular são:

a) a aquisição de conhecimentos teóricos para a compreensão dos vários paradigmas de Aprendizagem Automática, das metodologias e dos princípios da sua aplicação;

b) a aptidão para implementar estes algoritmos e, assim, resolver problemas práticos;

c) a competência para avaliar a adequação de cada método a cada caso particular e para avaliar criticamente os resultados obtidos.

O objetivo a) será atingido pela exposição da matéria, discussão e esclarecimento de dúvidas nas aulas teóricas, onde se abordará o fundamento teórico desta unidade curricular. Esta matéria será avaliada em testes individuais, onde os alunos têm um tempo limitado para mostrar o seu grau de assimilação e compreensão dos assuntos.

Os objetivos b) e c) serão atingidos principalmente pela componente prática da disciplina, se bem que sempre assente nas aulas teóricas, por meio das aulas práticas e do trabalho autónomo do aluno. As aulas práticas darão aos alunos oportunidade para aplicar os conhecimentos a casos concretos, experimentar as diferentes abordagens e examinar, em detalhe, os vários algoritmos em condições realistas. A componente prática será avaliada em dois trabalhos de grupo, avaliação esta que focará não só a implementação e aplicação prática dos algoritmos mas também a capacidade crítica de avaliação dos métodos escolhidos e dos resultados. Esta componente prática contribuirá também para a autonomia dos alunos.

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

The course objectives are:

- a) the acquisition of theoretical knowledge to allow the understanding of the different Machine Learning paradigms, methodologies and their principles of application;*
- b) the skill to implement these algorithms and, thus, solve practical problems;*
- c) the competence to evaluate the suitability of each method to each specific situation and for a critical evaluation of the results.*

Objective a) will be reached mainly through the theoretical lectures, which include exposition of the subject matter and discussion. These lectures will cover the theoretical foundation of the course and this knowledge will be evaluated in individual written tests, where the students will have limited time to show their success in assimilating and understanding the subjects.

Objectives b) and c) will be reached mainly through practice and the practical assignments, during the tutorial classes and the student's own work. Tutorial classes will give the students the opportunity to apply their knowledge to specific cases, experiment with different approaches and examine, in detail, the different algorithms in realistic conditions. This practical component will be evaluated in two group assignments, focusing on the implementation and practical application of the algorithms but also on the critical assessment of the methods and results. This component will contribute to give the student the autonomy and self-reliance needed for future applications and further studies in this field.

4.4.9. Bibliografia de consulta/existência obrigatória:

- *T. Mitchell. Machine Learning, McGraw-Hill, 1997.*
- *C. M. Bishop. Pattern Recognition and Machine Learning, Springer, 2006.*
- *E. Alpaydin. Introduction to Machine Learning, Second Edition, MIT Press, 2010.*
- *Stephen Marsland. Machine Learning: An Algorithmic Perspective CRC Press, 2011*

Mapa IV - Concorrência e Paralelismo

4.4.1.1. Designação da unidade curricular:

Concorrência e Paralelismo

4.4.1.1. Title of curricular unit:

Concurrency and Parallelism

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):
João Manuel dos Santos Lourenço (Regente) - T:28; PL:84

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):

Saber:

1. Compreender os conceitos de concorrência e paralelismo;
2. Conhecer os paradigmas utilizados no desenvolvimento de algoritmos concorrentes e paralelos;
3. Conhecer linguagens, bibliotecas e ferramentas utilizadas para desenvolver programas concorrentes e paralelos;
4. Conhecer os problemas de concorrência comuns e como os mitigar e evitar.

Saber Fazer:

5. Identificar e explorar oportunidades para concorrência e paralelização;
6. Particionar um problema em múltiplas tarefas para serem executadas num sistema paralelo;
7. Raciocinar sobre o comportamento de sistemas concorrentes e paralelos;
8. Construir sistemas concorrentes e paralelos corretos e eficientes;
9. Utilizar linguagens de programação e bibliotecas para desenvolver sistemas de software concorrentes e paralelos;
10. Utilizar ferramentas de programação para desenvolver aplicações concorrentes e paralelas;
11. Medir e prever o desempenho de sistemas paralelos.

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

To know:

1. Understand the concepts of concurrency and parallelism;
2. Know the paradigms used in the development of concurrent and parallel algorithms;
3. Know the languages, libraries and tools used in developing concurrent and parallel programs;
4. Know the common concurrency issues and how to mitigate and avoid them.

Know how to do:

5. Identify and explore opportunities for concurrency and parallelism;
6. Partition a problem into multiple tasks to be performed on a parallel system;
7. To reason about the behavior of concurrent and parallel systems;
8. Build correct and efficient concurrent and parallel systems;
9. Use programming languages and libraries to develop concurrent and parallel software systems;
10. Use programming tools in the development of concurrent and parallel applications.
11. Measure and predict the performance of parallel systems.

4.4.5. Conteúdos programáticos:

1. Programação paralela

O espectro dos problemas computacionais extremamente exigentes; problemas regulares e irregulares; estratégias para a decomposição de problemas e o seu mapeamento em padrões de programação; os modelos transaccional e map-reduce.

2. Arquiteturas paralelas

Taxonomia de Flynn; teoria do desempenho (incluindo leis de Amdahl e Gustafson).

3. Controlo de concorrência e sincronização

Competição e colaboração; atomicidade; linearização; monitores, locks; semáforos; barreiras; produtor-consumidor; locks de leitura e escrita; futuros, concorrência na prática em Java e C.

4. Safety e liveness

Safety vs. liveness; progresso; deadlock; prevenção, deteção e recuperação de deadlocks; livelocks; prevenção de livelocks; inversão de prioridade; herança de prioridade. Algoritmos lock-free.

5. O modelo transaccional

Operações compostas; transações (serialização), controlo de concorrência otimista (OCC) e memória transaccional.

6. Concorrência sem partilha de dados

Objetos ativos; troca de mensagens; atores.

4.4.5. Syllabus:

1. Parallel programming

The spectrum of high-demanding computational problems; regular and irregular problems; strategies for problem decomposition and their mapping to programming patterns; the transactional and map-reduce models.

2. Parallel architectures

Flynn's taxonomy; performance theory (including Amdahl's and Gustafson's laws).

3. Concurrency control and synchronization

Competition and collaboration; atomicity; linearization; monitors; locks; semaphores; barriers; producer-consumer; multi-reader single-writer locks; futures; concurrency in practice in Java and C.

4. Safety and liveness

Safety vs. liveness; progress; deadlock; deadlock prevention, avoidance, detection, and recovery; livelock; livelock avoidance; priority inversion; priority inheritance. Lock-free algorithms.

5. The transactional model

Composite operations; transactions (serializability), optimistic concurrency control (OCC) and transactional memory.

6. Concurrency without shared data

Active objects; message passing; actors.

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

Objetivos 1 e 7 são endereçados nos pontos 1 e 3 do programa.

Objetivo 2 é endereçado em todos os 5 pontos do programa.

Objetivos 3 e 4 são endereçados no ponto 3 do programa.

Objetivo 5, 9 e 10 são endereçados nos pontos 1, 3, 4 e 5 do programa.

Objetivo 6 é endereçado nos pontos 1, 4 e 5 do programa.

Objetivo 8 é endereçado nos pontos 2 e 3 do programa.

Objetivo 11 é endereçado nos pontos 3 do programa.

Todos estes objetivos são também endereçados na componente de formação prática, onde os alunos desenvolvem projetos de programação recorrendo às linguagens C e Java e a ferramentas para avaliação de desempenho, validação e correção de erros.

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

Objectives 1 and 7 are addressed in points 1 and 3 of the program.

Objective 2 is addressed in all 5 points of the program.

Objectives 3 and 4 are addressed in point 3 of the program.

Objective 5, 9 and 10 are addressed in points 1, 3, 4 and 5 of the program.

Objective 6 is addressed in points 1, 4 and 5 of the program.

Objective 8 is addressed in points 2 and 3 of the program.

Objective 11 is addressed in points 3 of the program.

All these objectives are also addressed in the practical training component, where students develop programming projects using C and Java languages and tools for performance evaluation, validation and error correction.

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 analisam, implementam e avaliam algoritmos concorrentes e paralelos. Algumas aulas práticas são dedicadas à realização dos projetos de programação.

Classificação final CF:

$$CF = 0.5 * CTP + 0.5 * CL + 0.05 * CP$$

— *Componente Teórico-Prática (CTP)* —

Obtém-se por exame individual.

— *Componente Laboratorial (CL)* —

Resulta da seguinte ponderação:

— *60% CG = classificação grupo no projeto.*

— *40% CI = classificação individual no projeto (definida pelo mérito do código e do relatório que descreve, avalia e analisa a solução desenvolvida).*

A CI é definida pelo mérito individual no processo de desenvolvimento do projeto.

— *Componente Participação (CP)* —

Definida pela quantidade e teor das participações nos fóruns Piazza e Slack.

4.4.7. Teaching methodologies (including students' assessment):

Teaching consists of exposing the syllabus in lectures and problem solving in laboratory practical classes. In the lab, students analyze, implement, and evaluate concurrent and parallel algorithms. Some practical classes are dedicated to the realization of programming projects.

Final classification CF:

$$CF = 0.5 * CTP + 0.5 * CL + 0.05 * CP$$

— *Theoretical-Practical Component (CTP)* —

Obtained by individual examination.

— *Laboratory Component (CL)* —

Results from the following weighting:

— *60% CG = group classification in the project*

— *40% CI = individual classification in the project (defined by the merit of the code and the report that describes, evaluates and analyzes the developed solution).*

CI is defined by individual merit in the project development process.

— *Participation Component (CP)* —

Defined by the number and content of participation in the Piazza and Slack forums.

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 endereça ao longo do semestre os objetivos conteúdo programáticos 1 a 5 e os objetivos 1 a 11.

O desenvolvimento de exercícios aplicados em aulas laboratoriais permitem aos alunos treinar os conhecimentos adquiridos na componente teórica e por estudo. O treino é desenvolvido no contexto dos objetivos “Saber Fazer” (5 a 11).

O desenvolvimento de um projeto de programação, nas aulas práticas e autonomamente, permite aos alunos consolidar os conhecimentos adquiridos nas aulas teóricas e estudo e treinados nas aulas laboratoriais.

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

The teaching methodology involves:

The theoretical exposition of the syllabus addresses throughout the semester the syllabus 1 to 5 and objectives 1 to 11.

The development of exercises applied in laboratory classes allow students to train the knowledge acquired in the theoretical component and by study. The training is developed in the context of the “Know-How” objectives (5 to 11).

The development of a programming project, in the practical and autonomous classes, allows the students to consolidate the knowledge acquired in the theoretical and study classes and trained in the laboratory classes.

4.4.9. Bibliografia de consulta/existência obrigatória:

1. McCool M., Arch M., Reinders J.; Structured Parallel Programming: Patterns for Efficient Computation; Morgan Kaufmann (2012); ISBN: 978-0-12-415993-8

2. Raynal M.; Concurrent Programming: Algorithms, Principles, and Foundations; Springer-Verlag Berlin Heidelberg (2013); ISBN: 978-3-642-32026-2

Mapa IV - Construção e Verificação de Software

4.4.1.1. Designação da unidade curricular:

Construção e Verificação de Software

4.4.1.1. Title of curricular unit:

Software Construction and Verification

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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:

João Ricardo Viegas da Costa Seco - T:28; PL:84

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

Saber:

Os alunos consolidam as aptidões de construção de desenvolvimento de software confiável, no contexto dos modernos sistemas, onde a concorrência e a segurança desempenham um papel crucial.

São cobertas as técnicas fundamentais de análise estática e de verificação de modelos, assim como princípios e técnicas de teste de software.

Saber Fazer

Usar métodos de programação rigorosos e técnicas de verificação para garantir a fiabilidade de programas concorrentes baseados em monitores e transações.

Desenvolver, em trabalho de grupo, o projecto de uma aplicação de média dimensão, estaticamente verificada e testada com um grau de cobertura razoável.

Usar asserções lógicas para especificar, verificar e raciocinar sobre a correcção de programas, assim como as ferramentas associadas.

Definir fluentemente especificações comportamentais (invariantes, pre-condições e pós-condições) para a implementação de módulos e suas interfaces.

Conceber e implementar planos de teste.

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

Knowledge:

Students consolidate their software construction skills for building trustworthy software, in the broad context of modern software systems, where concurrency and security are major concerns.

Basic static analysis and software model-checking techniques are also covered, as well as principles and techniques for software testing.

Knowledge Application:

Use programming methods and verification techniques to enforce safety of monitor based concurrent programs (using `java.util.concurrent`) and transactions.

Develop, in team work, a project of a statically verified and tested (with reasonable coverage) medium scale application.

Use logical assertions to specify, check, and reason about program correctness, and associated tools (DAFNY, VERIFAST, JIF).

Specify behavioral specifications (invariants, pre-conditions and post-conditions) for module implementations and their interfaces.

Develop test plans and implement them.

4.4.5. Conteúdos programáticos:**1. Construção Verificada de Software**

Métodos baseados em Asserções e Lógicas de Hoare e Separação. Inferência de Asserções. Tipos Abstratos e Tipos comportamentais; Invariantes de Representação. Interpretação Abstracta. Verificação de Modelos. Ferramentas associadas.

2. Teste de Software

Testes baseado em Modelos e Falhas. Seleção e Geração de Testes. Execução Simbólica. Ferramentas associadas.

3. Programação Concorrente

Partilha, Confinamento, Posse. Controle de Interferência. Verificação de código com monitores e semáforos usando invariantes de recursos. Controle de concorrência a partir de especificações comportamentais.

4. Exercícios de Desenvolvimento e Projecto Final

Sequência de desafios de programação. Uso de ferramentas (Dafny, JBoss, Verifast, SPIN; INFER)

4.4.5. Syllabus:**1. Verified Software Construction**

Assertion methods and Hoare and Separation Logic; Assertion Inference; Abstract and Behavioral types.

Representation Invariants. Abstract interpretation; Model-checking. Tools.

2. Software Testing *Model-based testing; Test selection and test generation; Fault-based testing. Symbolic execution; Automated testing. Tools.*

3. Concurrent Programming

Sharing, confinement, ownership. Control of interference. Reasoning about concurrent code with monitors and locks based on resource invariants. Construction of concurrency control code from behavioral specs.

4. Hands On Exercises / Final Project

Sequence of programming challenges, involving tool usage (Dafny, JBoss, Verifast, SPIN; INFER). Final (team work) project.

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

Esta UC desenvolve o ponto de vista de que a verificação deve ser integrada de forma coesa com os processos de construção de software, orientada pelo uso de ferramentas de análise de código. Os métodos de verificação de software baseados em lógica de Hoare e as suas extensões introduzem métodos de construção de software garantindo a sua correção a cada passo do desenvolvimento.

A utilização de linguagens de utilização corrente mostra a aplicação das teorias e ferramentas de verificação em situações de desenvolvimento de sistemas críticos. O primeiro resultado que se pretende é a aprendizagem de técnicas de especificação formal de software com lógica de primeira ordem (com a ferramenta Dafny) e lógica de

separação (Verifast) no sentido de verificar sistemas com memória alocada dinamicamente e concorrência. As mesmas técnicas de especificação podem ser aplicadas quer na verificação formal de correção como na verificação experimental através do desenho de testes funcionais.

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

This course develops the point of view from where software verification is fully integrated with the software construction process, guided by the use of static analysis tools. Correctness by construction is induced by the use of software verification methods based on Hoare logic and its extensions, which ensures the verification of the correctness conditions on every step of development process.

The use of mainstream languages illustrates the application of theoretical results to the actual development of critical systems. The first outcome that is sought is the learning of specification techniques using formal languages like first-order logic (using the Dafny tool) and separation logic (using Verifast). These tools are capable of verifying the correctness of systems with dynamic data structures and concurrency.

The same specification techniques are then applied in a scenario of experimental validation of software using functional testing.

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas da UC introduzem os diferentes tópicos com realce para técnicas de especificação de software usando lógica de primeira ordem e lógica de separação. A expressividade das lógicas é estudada no caso de programas imperativos e em especial no caso em que existe concorrência e estruturas de dados dinâmicas. Os métodos e as tecnologias são apresentadas de forma integrada para realçar as especificidades das linguagens de especificação e ferramentas utilizadas (Dafny e Verifast).

As aulas de laboratório incluem o desenvolvimento de três projetos de média dimensão, onde a correção funcional é formalmente verificada com ferramentas, ou validada através de testes funcionais.

Em regime normal, a componente teórico-prática composta por testes ou um exame e tem um peso de 70% e uma nota mínima de 9,5. A componente laboratorial tem um peso de 30% na nota final e uma nota mínima de 9,5 valores.

4.4.7. Teaching methodologies (including students' assessment):

Lectures in this course introduce the different topics with a focus in the software specification using first-order logic, Hoare logic, and Separation logic. The expressivity of the different logics is studied in the case of regular imperative programs, and especially in the case of concurrent programs and the use of dynamically allocated data structures. The Methods and technologies involved are presented in an integrated way to highlight the specific details of each specification languages and tools used (Dafny and Verifast).

Lab classes include the development of medium-sized projects where students develop applications that are formally verified with tools or validated through functional tests.

In a normal regime, the theoretical component comprises tests or an exam with a weight of 30% of the final grade and a minimum grade of 9.5 points. The laboratory component weights 30% of the final grade and has a minimum grade of 9,5 points.

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

Os alunos aprendem nesta UC a especificar programas utilizando lógica de primeira ordem, lógica de hoare e lógica de separação. A aplicação dos conhecimentos transmitidos nas aulas teóricas é feito através de pequenos exercícios nas aulas práticas e depois em projectos de média dimensão.

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

In this course, students learn how to specify the behaviour of programs by using first-order logic, Hoare logic and Separation logic. The application of knowledge is integrated through a series of classroom exercises in lab classes, and then expanded in projects of medium size.

4.4.9. Bibliografia de consulta/existência obrigatória:

Dafny Guide : <http://rise4fun.com/Dafny/tutorial/guide>

"Program Development In Java: Abstraction, Specification, And Object-Oriented Design". Liskov/Guttag; MIT Press.

Java Concurrency in Practice, Goetz et al. Addison-Wesley, 2006.

VeriFast for Java: A Tutorial Jan Smans, Bart Jacobs, and Frank Piessens

(<http://people.cs.kuleuven.be/~bart.jacobs/verifast/verifast-java-tutorial.pdf>).

Language Based Information Flow Security, A. Sabelfeld, A. C. Myers, 2004.

Several classical papers by Liskov, Hoare, Dijkstra, Brinch Hansen, Doug Lea, O'Hearn, Schneider.

Related course at MIT: <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-005-elements-of-software-construction-fall-2011/index.htm>

Mapa IV - Interação Pessoa-Máquina

4.4.1.1. Designação da unidade curricular:

Interação Pessoa-Máquina

4.4.1.1. Title of curricular unit:*Human-Computer Interaction***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:28; PL:28***4.4.1.6. ECTS:**

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Nuno Manuel Robalo Correia (sem horas de contacto)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Teresa Isabel Lopes Romão - T:28; PL:84***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Saber**Ter consciência da importância da interface como componente de um sistema interativo.**Compreender os factores humanos que condicionam a utilização dos sistemas interativos.**Compreender e utilizar novos paradigmas tecnológicos e metáforas de interacção.**Conhecer e aplicar os princípios de usabilidade.**Conhecer e aplicar diversas técnicas de prototipagem.**Conhecer e aplicar diversas técnicas de avaliação de interfaces.**Enquadrar IPM num Projeto de Engenharia.**Discutir a investigação recente no campo da interação pessoa-máquina (HCI)**Saber fazer**Analisar e especificar os requisitos dos utilizadores.**Desenhar, implementar e avaliar interfaces (Desenho iterativo).**Criar soluções inovadoras de resposta a problemas de interação.***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***Knowledge**Be aware of the importance of the interface as a component of an interactive system.**Understand the human factors which drive the usage of computer systems.**Understand novel paradigms for human-computer interaction**Know and apply usability principles.**Know and apply prototyping techniques.**Know and apply interfaces evaluation techniques.**Fit HCI in the engineering project.**Describe and discuss current research in the field of HCI.**Application**Analyse and specify users requirements.**Iterative design: Design, implement and evaluate user interfaces.**Develop creative capabilities to come up with innovative solutions for interaction problems.***4.4.5. Conteúdos programáticos:***1. Introdução - Human-Computer Interaction (HCI): o quê, para quê, quando?**2. Princípios de usabilidade*

3. *Características dos sistemas interativos*
4. *Fatores humanos e tecnológicos que condicionam a utilização de um sistema interativo*
5. *Desenho centrado no utilizador e processo de desenho iterativo de interfaces*
6. *Análise de tarefas e utilizadores*
7. *Sketching e prototipagem*
8. *Princípios de desenho de interação*
9. *Estilos de interação*
10. *Desenho gráfico*
11. *Paradigmas de interação*
12. *Técnicas de avaliação*
13. *Perspetivas futuras*

4.4.5. Syllabus:

1. *Introduction - Human-Computer Interaction (HCI): What? Why? When?*
2. *Usability principles*
3. *Characteristics of interactive systems*
4. *Human factors in the HCI*
5. *User centered design and iterative design process*
6. *User and task analysis*
7. *Sketching and prototyping*
8. *Interaction design principles*
9. *Interaction styles*
10. *Graphic design*
11. *Interaction paradigms*
12. *Evaluation techniques*
13. *Future perspectives*

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

O programa inclui o processo e os princípios de desenvolvimento de interfaces para sistemas interativos, focando diversas metodologias para as diferentes fases do processo: análise de requisitos, desenho, implementação e avaliação. São apresentados os fatores humanos e tecnológicos que condicionam a utilização de um sistema interativo e que determinam os princípios de usabilidade. É discutida a investigação recente no campo da interação pessoa-máquina (HCI) e analisadas as perspetivas futuras.

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

The program includes the process and principles of user interfaces development, focusing on different methodologies for the different iterative design process stages: user and task analysis, design, implementation and evaluation. Human factors and technological principles, which determine the use of interactive systems, are presented and the principles of usability are studied. Recent research in the field of human-computer interaction (HCI) is discussed and future perspectives are analysed.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas é apresentada a matéria, ilustrada com exemplos de aplicação. As aulas laboratoriais destinam-se à apresentação e realização de trabalhos práticos sobre a matéria apresentada na aulas teóricas.

A avaliação continua é constituída pela componente teórico-prática (2 testes escritos individuais, ao longo do semestre) e pela componente laboratorial (1 trabalho prático que corresponde a um projeto de desenvolvimento de uma interface para um sistema interativo).

Cálculo da nota final: $NF = 35\%T1 + 35\%T2 + 30\%TPratico$

Notas mínimas para aprovação: (média (T1; T2) ≥ 10) AND (TPratico ≥ 10)

Os alunos que obtiveram nota positiva na componente laboratorial e não obtiveram aprovação nos testes escritos poderão ainda realizar um exame, cuja nota substituirá a nota dos testes no cálculo da nota final.

4.4.7. Teaching methodologies (including students' assessment):

In the theoretical lectures the course content is presented, illustrated with application examples. The lab classes are intended for presentation and implementation of practical work regarding the content exposed during theoretical lectures.

The evaluation of the course consists of: a theoretical-practical component (2 mid-term written individual tests) and a lab or project component (1 practical work, which accounts for a project to develop the user interface for an interactive system).

Final_grade = $35\%T1 + 35\%T2 + 30\%Prat_Work$

Course approval requires the following minimal grades: (mean (T1; T2) ≥ 10) AND (Prat_Work ≥ 10)

Students approved in the lab component and not approved in the tests may also perform a final exam, which grade replaces the grades of the tests in the final grade formula.

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" e no programa da unidade curricular são apresentados e discutidos em detalhe nas aulas teóricas, sendo complementados através da bibliografia recomendada e complementar. Estes tópicos são apresentados e ilustrados através de exemplos de aplicação, recorrendo-se também a vídeos para melhor transmitir as características dinâmicas da interação com sistemas interativos.

As aulas laboratoriais destinam-se à apresentação e realização de trabalhos práticos sobre a matéria apresentada nas

aulas teóricas. Nestas aulas, os alunos podem pôr em prática os conhecimentos adquiridos nas aulas teóricas e através do material didático de apoio, de forma a atingir os objetivos da componente “Saber fazer”, através da realização de alguns exercícios práticos e, principalmente, de um projeto de desenvolvimento de uma interface para um sistema interativo. Este projeto tem diversas fases que incluem a análise de utilizadores e tarefas, desenho, prototipagem, implementação e avaliação, envolvendo duas iterações do processo iterativo de desenho. O projeto é efetuado em grupos de 3 ou 4 alunos, que correspondem a pequenas equipas de desenvolvimento em que todos contribuem para a geração de ideias, desenho e execução da interface final, simulando a execução de um projeto real e estimulando a criatividade, as competências para trabalhar em equipa, a capacidade de resolução de problemas e o desenvolvimento das aptidões técnicas de cada aluno. Nas fases de avaliação de usabilidade, esta é efetuada entre grupos, o que permite aos alunos praticarem vários métodos de avaliação e experimentarem a avaliação como membros da equipa de desenvolvimento, mas também da perspetiva do utilizador. Os alunos devem frequentar a maioria das aulas práticas onde trabalham em grupo e discutem ideias, problemas ou possíveis soluções com o docente. Cada fase do projeto tem uma data de entrega, na qual o grupo de alunos deve entregar um relatório escrito acerca dos resultados da correspondente fase. No final, cada grupo tem que apresentar ao docente o trabalho desenvolvido, justificando as opções de desenho e implementação tomadas durante o desenvolvimento do projeto. Os testes e o exame permitem avaliar os conhecimentos teóricos dos alunos. Esta componente de avaliação é escrita e individual, envolvendo também, por parte do aluno, capacidades de gestão de tempo e de comunicação escrita.

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

All the topics mentioned in the “Knowledge” objectives and in the course’s program are presented and discussed in detail in the theoretical classes and complemented by the recommended and complementary bibliography. These topics are presented and illustrated through application examples, also using videos to better convey the dynamic characteristics of the interaction with interactive systems.

The laboratory classes are intended for the presentation and execution of practical work on the subjects presented in the theoretical classes. In these classes, students can put into practice the knowledge acquired in the theoretical classes and through the didactic support material, in order to achieve the objectives of the “Know-how” component, through some practical exercises and, mainly, through a project involving the development of the user interface for an interactive system. This project has several phases that include user and task analysis, design, prototyping, implementation and evaluation, involving two iterations of the iterative design process. The project is done in groups of 3 or 4 students, which correspond to small development teams in which everyone contributes to the generation of ideas, design and execution of the final interface, simulating the execution of a real project and stimulating creativity, teamwork skills, problem solving skills and the development of each student’s technical competences. In the usability evaluation phases, the evaluation is carried out between groups, allowing students to practice various evaluation methods and to gain experience in usability evaluation as members of the development team, but also from the user’s perspective. Students should attend most practical classes where they work in groups and discuss ideas, problems or possible solutions with the teacher. Each phase of the project has a deadline by which each group of students must submit a written report on the results of the corresponding phase. In the end of the project, each group has to present their work, justifying the design and implementation options taken during the project development.

The tests and the exam allow to evaluate the theoretical knowledge of the students. This assessment component is written and individual, also involving time management and written communication skills.

4.4.9. Bibliografia de consulta/existência obrigatória:

Recomendada

- *Dix, A., Finlay, J., Abowd, G. and Beale, R., Human-Computer Interaction. Prentice Hall Europe, London, 2003.*
- *Norman, D., The Design of Everyday Things. MIT Press, 1998.*
- *Nielsen, J., Usability Engineering, Academic Press, 1993.*

Complementar

- *Mullet, K. and Sano, D., Designing Visual Interfaces, Prentice Hall, 1995.*
- *Gonçalves, D., Fonseca, M.J., and Campos, P., Introdução ao Design de Interfaces. FCA, 2017.*
- *Shneiderman, B., Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, Reading, Massachusetts, 1998.*
- *Moggridge, B., Designing Interactions. MIT Press, Massachusetts, 2007.*
- *Tufte, E., Envisioning Information, Cheshire, Connecticut Graphic Press, 2003.*

Mapa IV - Interpretação e Compilação de Linguagens

4.4.1.1. Designação da unidade curricular:

Interpretação e Compilação de Linguagens

4.4.1.1. Title of curricular unit:

Interpretation e Compilation of Programming Languages

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Luís Manuel Marques da Costa Caires - T:28; PL:84***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 o objetivo de transmitir conhecimentos sólidos na conceção e implementação de linguagens de programação, através do estudo de conceitos sintáticos, semânticos e pragmáticos fundamentais. Saber 1.Conhecer a arquitetura e técnicas usadas no desenho de interpretadores, compiladores e sistemas de tipos 2.Conhecer os elementos base das linguagens de programação e a sua semântica 3.Definir linguagens de programação por composição de elementos fundamentais Fazer 4.Representar e manipular a sintaxe abstrata de uma linguagem como dados de um programa transformador 5.Descrever a semântica de linguagens através de interpretadores, compiladores e verificadores de tipos 6.Conceber e implementar compiladores para uma máquina virtual industrial Soft-Skills 7.Raciocinar sobre sistemas complexos a vários níveis de abstração 8.Propor e concretizar soluções gerais com base em princípios

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

This course aims at teaching solid knowledge and concepts in the design and implementation of programming languages. This goal is reached by the systematic study of the syntax, semantics and pragmatics of programming languages. Knowledge 1.To know the architecture and techniques used in the design and implementation of interpreters, compilers and type systems 2.To know the essential components of the design of programming languages and corresponding semantics 3.To define programming languages by composition of base elements Skills 4.To define algorithms of the abstract representation of programs 5.To describe language semantics by interpreting, compiler and verification algorithms 6.To design and implement compiler procedures targeting concrete virtual machines Soft-Skills 7.To reason about complex systems at different levels of abstraction 8.To design general purpose designs based on first principles

4.4.5. Conteúdos programáticos:

A.Princípios 1.Sintaxe das Linguagens de Programação 2.Semântica das Linguagens de Programação 3.Níveis de Interpretação (Interpretação, compilação e sistemas de tipos) B.Interpretação de Programas 1.Valores e Expressões: Definições e Ambiente 2.Estado e Referências: Modelo ambiente-memória 3.Abstração funcional: funções de primeira classe e ordem superior 4.Definições recursivas e ambientes circulares 5.Valores estruturados: registos e valores recursivos 6.Objetos e classes: representação de objetos usando registos e fechos C.Sistemas de Tipos 1.Princípios, objetivos e limitações da análise estática 2.Sistemas de tipos simples 3.Algoritmos de verificação e inferência de tipos D.Compilação de Programas 1.Arquitetura de um compilador 2.Ambientes de suporte à execução (máquinas de registos, pilha de avaliação, pilha de chamada) 3.Tradução de código dirigida pela sintaxe 4.Geração de código para uma máquina virtual industrial (CLR,Java Virtual Machine)

4.4.5. Syllabus:

A.Principles 1.Programming Language Syntax 2.Programming Language Semantics 3.Interpretation levels, Principles of Language Processing and Compilation B.Program interpretation and Compilation 1.Expressions and values; Definitions and environments 2.State and references; memory-environment model 3.Functional abstraction; first-class and higher order functions 4.Recursive definitions 5.Structured values: records and recursive values 6.Objects and Classes: Objects represented with records and closures C.Type Systems 1.Principles, goals and limitations of static analysis methods 2.Simple type systems: functional types 3.Verification and inference algorithms D.Compilers and Runtime support systems 1.Compiler architecture 2.Runtime support systems 3.Syntax driven code transformation 4.Code generation to intermediate languages (CLR, Java Virtual Machine)

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

Os conteúdos programáticos cobrem de forma coerente os princípios e tecnologias associados à interpretação, compilação e verificação de tipos de linguagens de programação modernas.

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

The course syllabus covers principles and technologies for understanding and building interpreters compilers and type checkers for modern programming languages.

4.4.7. Metodologias de ensino (avaliação incluída):

A metodologia pedagógica proposta para esta unidade curricular centra-se na construção gradual de uma linguagem de programação. O ponto de partida é uma linguagem núcleo de expressões aritméticas, à qual se vão acrescentando novos elementos fundamentais numa espiral até se obter uma linguagem com funções de primeira classe e de ordem superior, capazes de exprimir as construções mais sofisticadas das linguagens de programação como classes, mixins ou closures. A avaliação decompõe-se em dois testes contribuindo cada um 30% para a nota final, o primeiro focado em aspetos básicos de definição de semântica de linguagens, e o segundo, na verificação através de sistemas de tipos. O restante da nota final (40%) é obtida através do desenvolvimento de um projeto de construção pragmática de um interpretador, compilador e sistema de tipos para uma linguagem dada.

4.4.7. Teaching methodologies (including students' assessment):

The teaching methodologies prescribed for this course is focused in the incremental construction of a programming language. The base language is a core expression language containing only arithmetic operators. The course follows in a spiral of new fundamental elements added to the language. The target language is a programming language with first-class and high order functions, capable of expressing more sophisticated constructs as classes, mixins, or closures. The evaluation comprises two tests, valued 30% in the final grade. The first test focuses on basic aspects of programming language semantics, the second is focused on type systems. The remainder of the final grade is obtained by developing a medium-sized project targeting a pragmatic programming language. The project comprises an interpreter, compiler and type system.

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

Esta unidade tem por objetivo transmitir um conjunto de princípios(A) abordados transversalmente, e que contribuem para atingir os objetivos 1,4 e 8. A definição de algoritmos interpretadores de forma incremental(B.1-6) é o fio condutor para atingir os objetivos 2, 3 e 5, e permite definir com precisão a semântica das construções nos vários níveis de interpretação(B,C,D) (semântica estática e semântica dinâmica) (objetivo 7). É colocada ênfase na composicionalidade dos elementos base(B), promovendo a compreensão de construções de análise mais complexa, comuns nas linguagens de programação de uso corrente (objetivo 8). Por outro lado, a exigência de construção de um projeto de média dimensão, promove o conhecimento profundo acerca da arquitetura de um interpretador (B), de um compilador(C.3), de um sistema de suporte à execução (D.2) assim cumprindo os objetivos 3,4 e 6, bem como qual o papel de um verificador baseado em sistemas de tipos, objetivo 5.

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

This course aims at transmitting a set of cross cutting principles(A) that contribute to reach goals 1, 4, and 8. The incremental definition of interpreter algorithms(B.1-6) constitutes a central line to reach goals 2,3,and 5, allowing to precisely define the semantics of the different language elements at different interpretation levels (B,C,D) - static semantics and dynamic semantics (goal 7). We focus on the compositionality of base elements(B) as a way of understanding more complex constructs which are common in mainstream programming languages (goal 8). On the other hand, the development of a middle-sized project promotes the deeper understanding of the architecture of an interpreter (B), compiler (C.3), or a runtime support system (D.2) thus contributing to goals 3,4, and 6, and the role of a typechecker algorithm, goal 5.

4.4.9. Bibliografia de consulta/existência obrigatória:

- "Concepts in Programming Languages", John C. Mitchell, Cambridge University Press. ISBN 0 521 78098 5 - "Essentials of Programming Languages", Daniel Friedman, Mitchell Wand, Christopher Haynes, MIT Press. - "Compiling for the .Net Common Language Runtime (Clr) (.NET Series)", John Gough, Prentice Hall PTR - "Modern Compiler Implementation in Java" Andrew W. Appel, Cambridge University Press - "The Study of Programming Languages", Ryan Stansifer, Prentice Hall International Edition.

Mapa IV - Modelação de Sistemas e Processos

4.4.1.1. Designação da unidade curricular:

Modelação de Sistemas e Processos

4.4.1.1. Title of curricular unit:

Systems and Process Modelling

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:28; PL:28***4.4.1.6. ECTS:**

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***João Baptista da Silva Araújo Junior - T:14; PL:56***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Miguel Carlos Pacheco Afonso Goulão - T:7**Vasco Miguel Moreira Amaral - T:7; PL:28***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 modelar processos organizacionais tais como contexto de sistemas de software e ainda desenvolvimento baseado em modelos e animação (ou, execução) de modelos de software. Em particular, os alunos devem:*

- *Compreender o papel dos modelos no desenvolvimento de software; modelos BPM e SysML; ferramentas de apoio à execução de modelos (Papyrus); técnicas para modelação de reutilização em larga escala.*
- *Ser capaz de usar eficazmente as ferramentas acima para especificar e executar modelos de software.*
- *Conhecer os conceitos e as técnicas de modelação de processos de negócio e sistemas.*

4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):*At the end of this curricular unit the student will have acquired knowledge, skills and competences that will allow him to model organizational processes such as context of software systems and also model-based development and animation (or, execution) of software models. In particular, students should:*

- *Understand the role of models in software development; BPM and SysML models; tools to support the execution of models (Papyrus); techniques for large-scale reuse modeling.*
- *Be able to effectively use the above tools to specify and run software models.*
- *Know the concepts and techniques of modeling business processes and systems.*

4.4.5. Conteúdos programáticos:*Introdução ao desenvolvimento baseado em modelo**Fundamentos do BPM*

- *Notação BPM*
- *Estilos de modelagem BMP*
- *Ferramenta BPM*

Diagramas de estrutura SYSML de

- *Definição de Blocos*
- *Blocos Internos*
- *Pacotes*

- *Paramétrico*
- *Requisitos*

Diagramas de comportamento SYSML de

- *Caso de Uso*
- *Atividade*
- *Máquina de Estado*
- *Sequência*

Processo de Engenharia de Sistemas

- *Captura de Requisitos, a visão e o domínio do sistema*
- *Capturando requisitos funcionais e suplementares*

Análise

- *Descrver uma solução de sistema conceitual - uso efetivo de diagramas de bloco, diagramas de sequência e statecharts*

- *Demonstrar como os requisitos são realizados com base na estrutura proposta*
- Design para criar hierarquia de sistema e / ou blocos de sistema independentes e reutilizáveis*
- *Modelando a estrutura física*
- *Preparando os requisitos para o próximo nível*
- *Rastreabilidade de modelagem*
- Modelagem para reutilização no grande*
- *Commonalities*
- *Variabilidade e gerenciamento de variabilidade*

4.4.5. Syllabus:

Introduction to model-based development

BPM fundamentals

- *BPM notation*
- *BPM modeling styles*
- *BPM tool*

SysML Structure Diagrams

- *Block Definition diagram*
- *Internal Block Diagram*
- *Package Diagram*
- *Parametric Diagram*
- *Requirement Diagram*

SysML Behavior Diagrams (Use Case Diagram, Activity Diagram, State Machine Diagram, Sequence Diagram)

System Engineering Process

- *Requirements Capture, the vision and system domain*
- *Capturing functional and supplementary requirements*

Analysis

- *Describing a conceptual system solution – effective use block definition, sequence and state chart diagrams*
- *Demonstrating how requirements are realized based on the proposed structure*

Design to creating system hierarchy and/or independent and reusable system blocks

- *Modeling physical structure*
- *Preparing the requirements for the next level*
- *Modeling traceability*

Modeling for reuse in the large

- *Commonalities*
- *Variabilities and variability management*

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

Inicialmente os conceitos de modelos de processos de negócio são apresentados, seguidos da notação BPMN juntamente com ferramentas de suporte. O sistema a ser desenvolvido e que faz parte de um processo de negócio será especificado em SysML, também com apoio de ferramentas que possibilitem a animação dos modelos especificados. Um conjunto de modelos SysML será ensinado cobrindo as características estruturais e dinâmicas do sistema e utilizado para análise e desenho do sistema. Por fim serão apresentadas técnicas específicas de reuso em larga escala.

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

Initially the concepts of business process models are presented, followed by BPMN notation along with support tools. The system to be developed that is part of a business process will be specified in SysML, also with tool support that enable the animation of the specified models. A set of SysML models will be taught covering the structural and dynamic characteristics of the system and used for system analysis and design. Finally, specific reuse techniques for large scale systems will be presented..

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas lecionam-se os conceitos teóricos; nas práticas, que decorrerão com recurso a ferramentas capazes de executar os modelos ensinados, aplicam-se as matérias lecionadas nas aulas anteriores.

No início das aulas os alunos receberão o planeamento para os tópicos do programa e a especificação do trabalho prático.

As aulas teóricas são dadas em sala equipada com um projetor e slides em ppt. As aulas práticas são realizadas em salas equipadas com computadores e projetor.

A avaliação consiste em: trabalho prático entregue em duas fases; dois testes. O trabalho é realizado em grupo e os testes individualmente.

A nota é uma média ponderada do trabalho prático (Fase I = 20%; Fase II = 30%), e das notas dos testes (Teste 1= 25%, Teste 2 = 25%).

Trabalho prático: Um trabalho prático obrigatório entregue em duas fases, valendo 20% (da nota final) na 1ª fase e 30% (da nota final) na 2ª fase. Para obter frequência, a média ponderada das duas fases deve ser superior ou igual a 9,5.

4.4.7. Teaching methodologies (including students' assessment):

Lectures teach the theoretical concepts and in labs, carried out using tools capable of executing the taught models, the subjects taught in previous classes are applied.

At the beginning of the class the students will receive the planning for the topics of the program and the specification of the practical work. A deadline is given for the delivery of the work.

Theoretical classes are given in a classroom equipped with a projector and ppt slides. The practical classes are held in

rooms equipped with computers and projector.

The assessment is composed of: a practical work delivered in 2 phases; 2 tests. Practical works and essays are done in groups and tests, individually.

The score is a weighted average of the practical work (Phase I = 20%; Phase II = 30%) and test score (Test 1= 25%, Test 2 = 25%).

Practical work: A practical work consists of 20% (of the final) in Phase 1 and 30% (of the final) in Phase 2. For frequency, the weighted average of the two phases should be ≥ 9.5 .

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

Nas aulas teóricas todos os tópicos do programa são apresentados, sempre ilustrados com exemplos inspirados em situações reais. Nas aulas práticas os conceitos introduzidos na teórica são consolidados com exercícios específicos. Nas práticas também é realizado o acompanhamento dos trabalhos a fim de avaliar o seu progresso.

O trabalho prático serve para os alunos praticarem os conceitos adquiridos de forma integrada, aplicada a um caso de estudo comum. Tanto o trabalho prático como o relatório de investigação fomentam o trabalho em equipa, enquanto que os testes/exames avaliam a performance individual.

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

In the lectures all topics of the program are presented, always illustrated with examples inspired by real situations. In practical classes the concepts introduced in the theory are consolidated with specific exercises. In practical classes it is also carried out monitoring of the practical work and essay to assess their progress.

The practical work is for students to practice the concepts acquired in an integrated way, applied to a common case study. Both the practical work and the research report foster teamwork, while the tests/exams assess individual performance.

4.4.9. Bibliografia de consulta/existência obrigatória:

Friedenthal, Sanford, Alan Moore, and Rick Steiner. A practical guide to SysML: the systems modeling language. Morgan Kaufmann, 2014.

Scheer, August-Wilhelm. ARIS—business process modeling. Springer Science & Business Media, 2012.

Pohl, Klaus, Günter Böckle, and Frank J. van Der Linden. Software product line engineering: foundations, principles and techniques. Springer Science & Business Media, 2005.

Mapa IV - Representação de Conhecimento e Sistemas de Raciocínio

4.4.1.1. Designação da unidade curricular:

Representação de Conhecimento e Sistemas de Raciocínio

4.4.1.1. Title of curricular unit:

Knowledge Representation and Reasoning Systems

4.4.1.2. Sigla da área científica em que se insere:

I

4.4.1.3. Duração:

Semestral

4.4.1.4. Horas de trabalho:

168

4.4.1.5. Horas de contacto:

T:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):*João Alexandre Carvalho Pinheiro Leite - T:28***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Joerg Matthias Knorr - PL:84***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Conhecimento:*

- *principais linguagens de representação de ontologias e os seus algoritmos de raciocínio.*
- *expressividade dessas linguagens versus a complexidade computacional do raciocínio.*
- *linguagens de representação de conhecimento de senso comum em IA.*
- *algoritmos de raciocínio sobre essas linguagens.*

Aptidões:

- *usar as linguagens de representação de conhecimento na modelação de BDs e em IA;*
- *aplicar os mecanismos de raciocínio para testes de correção de modelos de BDs e formulação de perguntas expressivas em BDs;*
- *aplicar os mecanismos de raciocínio sobre essas linguagens para derivação de consequências em IA;*
- *relacionar essas linguagens com problemas de integração de dados em BDs;*
- *aplicar essas linguagens na representação de problemas;*
- *utilizar sistemas que implementam mecanismos de raciocínio sobre essas linguagens.*

Competências:

- *Aplicação de conhecimentos teóricos formais em aplicações práticas.*
- *Estar ciente de trade-offs no desenho.*

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

- *Main languages for the representation of ontologies and their reasoning algorithms.*
- *Expressiveness of these languages versus the computational complexity of reasoning.*
- *Languages for representation of common sense knowledge in AI.*
- *Reasoning algorithms on these languages.*

Know-how:

- *Use knowledge representation languages for modelling DBs, and in AI;*
- *Apply reasoning mechanisms to test for the correctness of DB models and to formulate expressive DB queries.*
- *Apply reasoning mechanisms for knowledge representation languages to derive consequences in AI;*
- *Relate these knowledge representation languages with data integration problems in DBs;*
- *Use these knowledge representation languages for problem modelling and solving;*
- *Use systems that implement reasoning mechanisms for knowledge representation languages.*

Soft-Skills:

- *Apply formal theoretical knowledge in real-world applications;*
- *Become aware of design trade-offs.*

4.4.5. Conteúdos programáticos:

A UC tem uma primeira parte que se foca em linguagens para especificação de domínios, e em algoritmos para raciocínio nestas linguagens. Na segunda parte, a UC foca em linguagens e sistemas para representação de conhecimento em problemas típicos da área de IA, nomeadamente em linguagens para representação de conhecimento de senso comum, e em problemas de satisfação, escalonamento e planeamento.

*1. Introdução à Representação de Conhecimento e Raciocínio**2. Sistemas baseados em Ontologias**2.1. Modelação de informação através de Ontologias**2.2. Linguagens de ontologias (Lógicas de Descrição)**2.3. Resposta a consultas a Bases de Dados e Ontologias**2.4. Acesso a dados baseado em Ontologias**2.5. Integração de dados baseada em Ontologias**2.6. Raciocínio em linguagens de Ontologias**3. Sistemas baseados em Regras**3.1. Raciocínio não-monotónico**3.2. Datalog**3.3. Acesso a dados baseado em Datalog**3.4. Integração de dados baseada em Datalog**3.5. Programação por Conjuntos de Resposta***4.4.5. Syllabus:**

The course has a first part, which focuses on languages for domain specification, and reasoning algorithms for these languages. In the second part, the course focuses on languages and systems for knowledge representation in typical problems of AI, particularly in languages for the representation of commonsense knowledge, and in problems of satisfaction, scheduling and planning.

*1. Introduction to Knowledge Representation and Reasoning**2. Ontology Based Systems*

- 2.1. Modeling information through ontologies
- 2.2. Ontology Languages (Description Logics)
- 2.3. Query answering over databases and ontologies
- 2.4. Ontology based data access
- 2.5. Ontology based data integration
- 2.6. Reasoning in Ontology Languages
- 3. Rule Based Systems
 - 3.1. Non-monotonic Reasoning
 - 3.2. Datalog
 - 3.3. Datalog based data access
 - 3.4. Datalog based data integration
 - 3.5. Answer-Set Programming

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

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

As aptidões objetivo da unidade curricular decorrem da exercitação do programa em atividades práticas laboratoriais. Todo o conteúdo programático contribui para o objetivo relacionado com a escolha e utilização de formalismos adequados à representação de conhecimento em áreas de aplicação práticas.

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

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

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

The know-how learning outcomes stem from the exercise of the program in lab sessions.

The entire syllabus contributes to the objective related to the choice and employment of adequate knowledge representation formalisms to represent knowledge in practical applicaiton areas.

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

4.4.7. Metodologias de ensino (avaliação incluída):

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

Nas aulas práticas são resolvidos problemas, feitas experiências em laboratório de computadores, e (parcialmente) desenvolvidos os trabalhos práticos.

A avaliação de conhecimentos inclui:

- 2 testes individuais teóricos (ou um exame de recurso), onde é avaliado o conhecimento que os alunos adquiriram dos conceitos e características das técnicas leccionadas (50%) (nota mínima na componente: 9,5 val.).

- 2 trabalhos práticos, elaborados em grupos de 2 estudantes, que consistem: a) no desenvolvimento de uma ontologia e sua utilização para acesso a dados e b) na modelação de um problema usando Programação por Conjuntos de Resposta (50%).

4.4.7. Teaching methodologies (including students' assessment):

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

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

The evaluation includes:

- 2 individual tests (or an exam) where the knowledge acquired by the students on the concepts and techniques is assessed (50%) (minimum grade in componente: 9.5 val.).

- 2 practical projects, developed in groups of 2 students, consisting of a) the development of an ontology and its use for data access and b) the modelling of a problem using Answer-Set Programming (50%).

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

Os objetivos da unidade curricular identificam a) a aquisição de conhecimento sobre técnicas e mecanismos de representação de conhecimento e raciocínio; b) a aplicação prática dos conhecimentos adquiridos e utilização de ferramentas existentes.

O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição de um conjunto de formalismos de representação de conhecimento e raciocínio.

O objetivo b) é fundamentalmente atingido nas aulas práticas através da aprendizagem e utilização das ferramentas de raciocínio existentes, seja para experimentar os vários conceitos aprendidos, seja para os aplicar na modelação e implementação dos vários trabalhos.

As matérias relativas ao objetivo a) são avaliadas no teste individual. As matérias relativas ao objetivo b) são avaliadas nos trabalhos práticos.

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

The objectives of the course identify a) the acquisition of knowledge about techniques and mechanisms for knowledge representation and reasoning; b) the practical application of acquired knowledge, and usage of existing tools.

Objective a) is mainly achieved in lectures with the presentation of a set of formalisms for knowledge representation and reasoning.

Objective b) is mainly achieved in practical classes through the learning and usage of existing reasoning tools, be it to experiment with the various learnt concepts, be it to apply them in modelling and implementation of the various projects.

Matters relating to the objectives a) are evaluated through individual tests. Matters relating to objective b) are evaluated through the practical projects.

4.4.9. Bibliografia de consulta/existência obrigatória:

Livros de Texto / Text Books

• *Knowledge Representation and Reasoning by Ronald Brachman & Hector Levesque, Morgan Kaufmann 2004.*

• *Answer Set Solving in Practice by M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool, 2012.*

• *Handbook of Knowledge Representation edited by Frank van Harmelen, Vladimir Lifschitz and Bruce Porter, Elsevier 2007.*

• *The Description Logic Handbook: Theory, Implementation, and Applications. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider. Cambridge University Press, 2003.*

Mapa IV - Sistemas de Bases de Dados

4.4.1.1. Designação da unidade curricular:

Sistemas de Bases de Dados

4.4.1.1. Title of curricular unit:

Databases 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:28; PL:28

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 Júlio Alves Alferes - T:28; PL:28

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

Joaquim Francisco Ferreira da Silva - PL:56

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

Conhecimento:

Conceitos fundamentais para o funcionamento de um sistema de bases de dados.

Noções de indexação e hashing em bases de dados.

Processamento e otimização de perguntas em bases de dados relacionais.

Concorrência e processamento de transações em bases de dados.

Noções básicas de bases de dados distribuídas.

Aptidões:

Otimizar bases de dados para utilizações específicas.

*Otimizar o processamento de perguntas, e utilizar as ferramentas comuns em SGBD para esse propósito.
 Usar os mecanismos de controlo de transações em bases de dados.
 Usar um SGBD para uma base de dados distribuída
 Utilização avançada do SQL e linguagens procedimentais com SQL.
 Competências:
 Capacidade de trabalho em equipa.
 Capacidade de avaliar soluções com base em resultados experimentais.
 Capacidade de avaliação crítica.*

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

Knowledge:

Fundamental concepts in database management systems.

Indexing and hashing in databases

Query processing and optimization in relational databases.

Concurrency and transaction processing in databases.

Basic notions of distributed databases.

Skills:

Optimize databases for specific uses.

Optimize queries processing, using common tools in DBMSs for such purpose.

Use of control mechanisms for concurrency

Use of DBMSs for distributed databases.

Advanced use of SQL and associated procedural languages.

Competences:

Team work.

Capacity to evaluate a solution based on experimental results.

Critical evaluation.

4.4.5. Conteúdos programáticos:

Implementação de SGBDs

1.1.Armazenamento e estrutura de ficheiros

1.2.Indexação

1.3.Hashing estático e dinâmico

Processamento e otimização de perguntas em SQL.

2.1.Processamento: esquema geral; algoritmos para operadores de álgebra relacional; composição de operadores por materialização e pipelining; algoritmos paralelos

2.2.Otimização: medidas de custo; estimativas associadas a planos de avaliação

Concorrência e processamento de transações

3.1.Introdução ao controlo de concorrência em bases de dados

3.2.Processamento de transações ACID e níveis de isolamento

3.3.Protocolos baseados em locks e em time-stamps

3.4.Protocolos multi-versão

3.5.Esquemas de recuperação em bases de dados

Fundamentos de Arquitecturas de Sistemas de Bases de Dados

4.1.Bases de dados centralizadas e cliente-servidor

4.2.Introdução às bases de dados distribuídas: homogéneas e heterogéneas; processamento de perguntas em bases de dados homogéneas; transações em bases de dados distribuídas

4.4.5. Syllabus:

DBMSs implementation

1.1.Storage and file structure

1.2.Indexing

1.3.Static and Dynamic Hashing.

SQL query processing and optimization

2.1.Processing: general schema; algorithms for relational algebra operators; operators composition – materialisation and pipelining; parallel algorithms

2.2.Optimization: cost measures; estimation of evaluation plans

Concurrency and transaction processing

3.1.Introduction to concurrency control in databases

3.2.Transaction processing and isolation levels

3.3.Lock and time-stamp based protocols

3.4.Multi-version protocols

3.5.Recovery in databases

Fundamentals of DBMS architecture

4.1.Centralised and client-server databases

4.2.Introduction to distributed databases: homogeneous and heterogeneous; query processing in distributed databases; transactions in distributed databases

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

O programa cobre, praticamente ponto a ponto, os vários conceitos mencionados nos objetivos de aprendizagem (conhecimento).

Os vários conceitos abordados no programa constituem um todo coerente para uma melhor compreensão do funcionamento de sistemas de gestão de bases de dados, sendo que tal compreensão é instrumental para que consigam fazer um melhor uso, e mais eficiente, de tais sistema, em especial quando se tratam grande quantidades de dados,.

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

The syllabus cover, almost item by item, all the concept mentioned in the objectives of the course.

The various concepts make a coherent body for a better understanding of the way database systems work. This understanding is instrumental for making a better, and more efficient, use of such systems, especially when implementing big databases.

4.4.7. Metodologias de ensino (avaliação incluída):

Como metodologia de ensino há aulas teóricas, onde são expostos os conceitos, e aulas práticas onde os estudantes fazem exercícios e avaliam, em sistemas de bases de dados (especialmente no Oracle), como é que os vários conceitos são materializados, e como é que tal influência a performance e usabilidade do sistema.

Quanto à avaliação, a nota final (NF) é dada por

$$NF = 0,75*NT + 0,25*NP,$$

sendo necessário pelo menos 10 valores na nota teórica (NT) para aprovação (para além de média final positiva). Todas as notas são de 0 a 20, arredondadas à unidade.

NT = (T1+T2)/2 (i.e. 2 testes). Alternativamente ou como melhoria, NT também pode ser obtida por exame de recurso.

NP resulta da realização, apresentação e discussão de um trabalho (em grupos de 3), e da realização individual de um relatório de avaliação de um outro grupo. Uma eventual nota prática do ano anterior é válida para esta edição (não realizando o trabalho). Alunos com nota prática mais antiga devem contactar o regente.

4.4.7. Teaching methodologies (including students' assessment):

The teaching methodology includes lectures, where the various concepts are discussed, and labs where student do exercises e evaluate, in existing database system (mainly Oracle), how the various concepts are materialised, and how that influences the performance and usability of the system.

Regarding the evaluation, 2 tests (or exam) yielding TG grade, and project yielding PG, for a final grade (FG) given by

$$FG = 0.75*TG + 0.25*PG.$$

For approval, both $FG \geq 10$ and $TG \geq 10$ are required.

Project consists of a group (of 3 students) assignment (including oral presentation) and of an individual evaluation report of another group's project.

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

A metodologia de ensino, combinado exposição e discussão de conceitos, com experimentação em sistemas reais, garante a obtenção dos objetivos da UO.

A avaliação, incluindo um trabalho prático, permite que os estudantes vejam como os conceitos funcionam na prática.

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

The teaching methodology, combining lectures where the concepts are discussed with experimentation in real systems, guarantees that the goal of the course are achieved.

The evaluation, inclding a practical project, ensure that the students understand how the concepts work in practice.

4.4.9. Bibliografia de consulta/existência obrigatória:

Text Books

- Database System Concepts. A. Silberschatz, H. Korth and S. Sudarshan, McGraw-Hill, 7th Edition, 2019

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

Manuals

Oracle 18c Documentation

Mapa IV - Sistemas de Computação em Cloud

4.4.1.1. Designação da unidade curricular:

Sistemas de Computação em Cloud

4.4.1.1. Title of curricular unit:
Cloud Computing 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:28; PL:28

4.4.1.6. ECTS:
6

4.4.1.7. Observações:
Optativa

4.4.1.7. Observations:
Optional

4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):
Nuno Manuel Ribeiro Preguiça - T:28; PL:84

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):
Conhecimentos

- * Compreender os fundamentos dos sistemas de computação em Cloud*
- * Compreender os diferentes modelos de serviço Cloud (IaaS, PaaS, SaaS). Exemplos, sua realização e características.*
- * Conhecer os centros de dados, as infraestruturas e compreender as restrições impostas.*
- * Conhecer os serviços disponíveis nas plataformas cloud, como funcionam e como podem ser usados.*
- * Conhecer as técnicas de virtualização de recursos.*

Aptidões e competências

- * Saber identificar as classes de aplicações que beneficiam dos sistemas de computação de Cloud.*
- * Saber criar soluções que tirem partido das características dos serviços de Cloud e respetivos recursos.*
- * Saber avaliar soluções desenvolvidas em plataformas Cloud.*

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

Knowledge

- * Understand the fundamentals of Cloud computer systems.*
- * Learn the different cloud service models (IaaS, PaaS, SaaS). Examples, characteristics and implementation.*
- * Know the data center infrastructures and the restrictions imposes.*
- * Know the services available in cloud platforms, how they work and can be used.*
- * Know resource virtualization techniques.*

Skills and competences

- * Identify the classes of applications that benefit from Cloud computing systems.*
- * Create solutions that take advantage of Cloud services and resources.*
- * Evaluate solutions developed for Cloud platforms.*

4.4.5. Conteúdos programáticos:

1. Introdução

1.1 Desafios e motivação

1.2 Modelos

1.3 Elementos dum sistema cloud

2. Serviços de armazenamento

2.1 Armazenamento de blocos e sistemas de ficheiros

2.2 Bases de dados Cloud: repositórios chave-valor e bases de dados

3. Serviços de caching

3.1 Cache da aplicação

3.2 Content-distribution networks

- 4. *Serverless computing*
- 4.1 *Suporte para micro-serviços*
- 4.2 *Function-as-a-Service*
- 5. *Processamento de dados*
- 5.1 *Serviços de filas de espera*
- 5.2 *Serviços de processamento em batch*
- 5.3 *Introdução ao processamento em stream*
- 6. *IaaS*
- 6.1 *Virtualização*
- 6.2 *Containers*
- 6.3 *Implantação em agregados (Kubernetes)*
- 7. *Redes de computadores*
- 7.1 *Redes virtuais*
- 8. *Segurança*
- 8.1 *Serviços de autenticação*
- 8.2 *Privacidade e segurança*
- 9. *Infraestrutura*
- 9.1 *O centro de dados físico*
- 9.2 *Arquiteturas avançadas: cloud híbrida*
- 9.3 *Computação na edge*

4.4.5. Syllabus:

- 1. *Introduction*
- 1.1 *Challenges and motivation*
- 1.2 *Models*
- 1.3 *Building blocks*
- 2. *Storage services*
- 2.1 *Block storage and filesystems*
- 2.2 *Cloud databases: key-value stores and databases*
- 3. *Caching services*
- 3.1 *Application caches at the DCs*
- 3.2 *Content-distribution networks*
- 4. *Towards serverless computing*
- 4.1 *Support for micro-services*
- 4.2 *Function-as-a-Service*
- 5. *Processing data*
- 5.1 *Queuing services*
- 5.2 *Batch processing services*
- 5.3 *Intro to stream processing services*
- 6. *IaaS*
- 6.1 *Virtualization*
- 6.2 *Containers*
- 6.3 *Deployment in clusters (Kubernetes)*
- 7. *Networking*
- 7.1 *Virtual networks*
- 8. *Security*
- 8.1 *Authentication services*
- 8.2 *Privacy and security*
- 9. *Infrastructure*
- 9.1 *The physical data center*
- 9.2 *Advanced cloud architectures: hybrid cloud*
- 9.3 *Edge computing*

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 aplicações que explorem as potencialidade das plataformas cloud. Para atingir esse objetivo, a UC começa por apresentar os principais desafios que se colocam nos sistemas de cloud, e modelos alternativos. De seguida, estudam-se os principais serviços disponíveis nas plataformas de cloud (armazenamento, caching, serverless, processamento). O programa prossegue com o estudo dos mecanismos de virtualização (IaaS, redes), seguido duma breve introdução aos serviços de segurança e estudo das infraestruturas cloud.

Na componente prática, os conceitos estudados na parte teórica são colocados em prática desenvolvendo de forma incremental, uma aplicação que usa serviços duma plataforma cloud e dum trabalho que se foca na implantação duma aplicação usando recursos cloud.

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

The goal of this course is to provide the knowledge and competences to develop applications that explore the potential of cloud platforms. To reach this goal, the course starts by presenting the main challenges posed to cloud systems and the alternative cloud models. The course follows by introducing the key services available in cloud platforms (storage, caching, serverless, processing). The syllabus proceeds with the study of virtualization mechanisms (IaaS and networks), followed by a brief introduction to security services and a study of cloud infrastructures.

In the labs, the topics studied in the lectures will be exercised in the development of an application that uses the services of a cloud platform and in a project that focus on the deployment of an application using cloud resources.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas explicam-se e discutem-se os tópicos do programa da cadeira. A abordagem consiste em apresentar, para cada tópico, a motivação/problema e soluções. Nesta apresentação procura-se discutir também as restrições que levaram a uma dada solução e as possíveis alternativas. Adicionalmente, apresentam-se exemplos de como usar as soluções apresentadas para no desenvolvimento de aplicações.

As aulas práticas acompanham os projetos da cadeira, fornecendo aos alunos a experiência sobre como é que os conceitos introduzidos na cadeira podem ser aplicados na prática. Desta forma, nas aulas práticas apresentam-se exemplos dos vários tópicos abordados nas teóricas e faz-se a ligação destes com os trabalhos que os alunos realizam ao longo da disciplina.

Avaliação: dois testes (25% cada), projeto principal (40%), projeto 2 (10%).

4.4.7. Teaching methodologies (including students' assessment):

In lectures the various topics of the syllabus are explained and discussed. The approach consists in presenting, for each topic, the motivation/problem and solutions. This presentation also discusses the constraints that led to a given solution and possible alternatives. In addition, examples are given of how to use the solutions presented for application development.

Laboratory classes follow the course projects closely, giving students the experience of how the concepts introduced in lectures can be applied in practice. Thus, lab class present examples of the various topics, making a connection to its use in the course projects.

Assesment: two tests (25% each), main project (40%), project 2 (10%).

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 alcançar os objetivos da disciplinas, em particular fornecer um conjunto de conhecimentos e competências que permitam ao alunos serem capazes de desenvolver aplicações que exploram as potencialidade das plataformas cloud.

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.

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

The teaching methodologies aim to achieve the objectives of the course, in particular to provide a set of knowledge and competences that allow students to be able to develop applications that exploit the potential of cloud platforms.

This is achieved by exposing the fundamental concepts and techniques in lectures and their consolidation and implementation in lab classes, by presenting examples and developing two projects.

4.4.9. Bibliografia de consulta/existência obrigatória:

Cloud Computing: Theory and Practice, Dan C. Marinescu, Morgan Kaufman-Elsevier, 2013.

The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Luiz André Barroso, Jimmy Clídaras, Urs Hölzle, Synthesis Lectures on Computer Architecture, Morgan & Claypool Publishers, 2013.

Selected conference and/or journal papers.

Mapa IV - Segurança de Redes e Sistemas de Computadores

4.4.1.1. Designação da unidade curricular:

Segurança de Redes e Sistemas de Computadores

4.4.1.1. Title of curricular unit:

Network and Computer Systems Security

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:28; PL:28***4.4.1.6. ECTS:**

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***José Augusto Legatheaux Martins (sem horas de contacto)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Henrique João Lopes Domingos - T:28; PL:84***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Conhecimentos**- Frameworks de segurança, normalização de serviços de segurança e modelos de segurança para sistemas e redes de computadores**- Métodos, ferramentas e algoritmos criptográficos**- Protocolos e serviços de segurança na pilha TCP/IP e aplicações Internet**- Segurança do Sistema de Operação e Bases de Dados**- Sistemas para computação confiável**Aptidões e competências práticas**- Programação com ferramentas criptográficas**- Desenho e implementação de protocolos e serviços de segurança para redes de computadores e sistemas distribuídos**- Implementação de propriedades de segurança para aplicações Internet**- Proteção de recursos ao nível do sistema de operação**- Conceção e implementação de protocolos seguros de acesso a dados e proteção de repositórios de gestão de dados**- Desenho e implementação de soluções de segurança para sistemas distribuídos de grande escala (Internet) bem como de proteção de privacidade de dados e operações.***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***Knowledge**- Security frameworks, security standards and security models for computer systems and networks**- Cryptographic methods, tools and algorithms**- TCP/IP security services and protocols and Internet security**- Operating Systems Security and Virtualization Techniques**- Trusted Computing and Trusted Software Execution Environments**Practical skills**- Programming with cryptographic tools and techniques**- Design and implementation of security protocols for distributed systems and internetworked applications and services**- Implementation and operation of security services and mechanisms in Operating Systems**- Design and implementation of secure data-management systems and data-access protocols**- Design and implementation of security properties and services for large scale distributed systems, including data-privacy guarantees and privacy-enhanced operation.***4.4.5. Conteúdos programáticos:***- Introdução**- Fundamentos de segurança em sistemas e redes de computadores**- Fundamentos, métodos e ferramentas da criptografia computacional aplicada**- Sistemas, protocolos e serviços de autenticação**- Serviços de controlo de acesso: modelos, políticas e mecanismos**- Serviços, protocolos e padrões de segurança na pilha TCP/IP**- Segurança de redes e canais de acesso (LAN, WAN)**- Segurança de sistemas: defesa de intrusões, defesa de perímetro e computação confiável.***4.4.5. Syllabus:**

- *Introduction*
- *Foundations and security models for computer systems and network security; study of security standard frameworks*
- *Applied cryptography and cryptographic tools and methods*
- *Authentication systems, services and protocols*
- *Access control services: policy models and mechanisms*
- *TCP/IP security stack: services, protocols and standards*
- *Network access control and LAN/WLAN security services*
- *Systems security: intrusion detection, trusted computing and perimeter defense systems.*

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

O programa da UC transmite conhecimentos e aptidões para conceção de mecanismos e serviços de segurança para sistemas e redes de computadores. As valências transmitidas consolidam uma base de conhecimento rigoroso e estruturante para modelação e conceção de mecanismos e serviços de segurança para sistemas distribuídos, com particular destaque para ambientes TCP/IP.

Os fundamentos teóricos cobrem modelos, métodos, bem como técnicas de conceção e verificação de propriedades de segurança, a partir: (i) da definição rigorosa do modelo de adversário (ii) do estudo de protocolos normalizados e seus mecanismos de conceção; (iii) da capacidade de delimitação da base de confiança de um sistema computacional; e (iv) do estudo dos serviços de segurança ao nível do sistema de operação e suporte de aplicações.

Nas aulas práticas desenvolvem-se as técnicas de implementação e ensaio experimental, com exercícios de demonstração, programação e realização de trabalhos de média complexidade.

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

This course, as a consolidation course in the MSc curricula, aims to provide students with solid foundations, knowledge base and know-how, allowing the development of secure internetworking applications and services, as well as, for designing large-scale distributed systems supported by the Internet.

The course provides theoretical foundations and a practical survey of network security protocols, applications and standards, including security solutions for the Internet and corporate networks and on related standards that have been widely deployed and are inspirational approaches to design new security protocols for distributed systems.

In labs, the lectures are exercised in the development of work-assignments illustrating the theoretical foundations, and evaluation projects with increasing complexity, as well as, for the experimental assessment and validation of security services and mechanisms for distributed computing systems.

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas da unidade curricular são dadas em língua inglesa, sendo todos os materiais também disponibilizados em língua inglesa.

A unidade curricular é composta por aulas teóricas, onde são transmitidos e discutidos os conceitos fundamentais e aulas práticas, desenvolvidas em ambiente laboratorial.

Nas aulas práticas são feitas apresentações de técnicas de segurança, podendo envolver demonstrações dessas técnicas (de acordo com o programa das aulas práticas) e realizados exercícios básicos relacionados com essas técnicas. Algumas aulas práticas serão dedicadas aos trabalhos práticos de avaliação a realizar pelos alunos, incluindo o esclarecimento de dúvidas, apresentação ou demonstração de resultados bem como discussão de possíveis soluções.

Avaliação: dois testes (intermédio: 25%; final: 35%), dois mini-projetos (20% cada).

4.4.7. Teaching methodologies (including students' assessment):

The course is prepared to be taught in English - all materials are in English.

The course is organized in lectures for presenting and discussing foundations, concepts, principles, paradigms, techniques or algorithms.

Labs are organized for presenting computer and network security techniques (following the program), involving the demonstration of such techniques or related components. Some sessions are planned for discussing practical solutions on proposed problems, as well as support for implementation of the assessment projects or work-assignments.

Assessment: two tests (midterm: 25%, final: 35%), two mini-projects (20% each).

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

A UC transmite conhecimentos e aptidões para conceção de mecanismos e serviços de segurança para sistemas e redes de computadores. As valências adquiridas estabelecem e consolidam uma base de conhecimento estruturante para desenho e implementação de sistemas distribuídos seguros e suas aplicações, operando em redes TCP/IP e na Internet.

As aulas teóricas transmitem fundamentos teóricos sobre mecanismos específicos, modelos, métodos e técnicas de concepção para implementação e verificação de propriedades de segurança com base (i) na definição rigorosa de modelos de adversário a partir de frameworks normalizadas de tipologias de ataques (ii) e da identificação e delimitação da base de confiança de um sistema.

Nas aulas práticas colocam-se em prática as técnicas de implementação e ensaio experimental, com exercícios de programação, utilização de ferramentas e realização de trabalhos de avaliação de média complexidade.

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

In this course, student will learn the fundamental principles including design and operational principles of computer and network security, by studying attacks on computer systems, network protocols, and internet-based or web-enabled applications and services. Students will learn the attacks taxonomy and how to prevent and detect them by

designing proper security services and mechanisms to protect computer systems and distributed applications. The course covers the study of relevant security protocols in the TCP/IP stack, from the data-link level to the application-level, including the design principles and the operation issues.

Lectures explore the theoretical foundations of security frameworks for security standards, specific security mechanisms, security services and protocols, as well as development methods and techniques used to design and implement secure internetworked applications and services for TCP/IP networks and for operation in the Internet. The course has a relevant component in "learning by doing" or "hands-on-approach" and requires students to conduct a series of lab exercises and in the development of evaluation practical work. Through those labs and mini-projects, students can enhance their understanding of the theoretical principles, and be able to apply the acquired knowledge to address real problems. After completion of the course, students should be able to possess the following skills:

- *Be able to explain security principles, properties, mechanisms and services*
- *Be able to define adversary models or threat-model conditions to design security services and mechanisms*
- *Be able to explain how various attacks work and possible protection mechanisms, by design.*
- *Be able to describe and generalize possible vulnerabilities in TCP/IP protocols and software vulnerabilities.*
- *Be able to understand the design principles and operation of various security protocols in the TCP/IP stack, at different approach levels.*
- *Be able to analyze and evaluate software system*

4.4.9. Bibliografia de consulta/existência obrigatória:

Main references

- *W. Stallings, Network Security Essentials - Applications and Services, Pearson, 6/E, 2017*
- *W. Stallings, L. Brown, Computer Security: Principles and Practice, Pearson 4/E, 2014*

Additional References

- *W. Stallings, Cryptography and Network Security - Principles and Practice, Pearson 7/E, 2017*
- *D. Gollmann, Computer Security, 3rd Ed, Wiley, 2011*

Other References

- *B. Schneier, Applied Cryptography, 1996, Wiley*
- *A. Zúquete, Segurança em Redes Informáticas, 5ª Ed., 2018, Ed. FCA*
- *M. Correia, P. Sousa, Segurança no Softwarem, 2ª Ed. , 2017 Ed. FCA*

Mapa IV - Algoritmos e Sistemas Distribuídos

4.4.1.1. Designação da unidade curricular:

Algoritmos e Sistemas Distribuídos

4.4.1.1. Title of curricular unit:

Algorithms and 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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Nuno Manuel Ribeiro Preguiça (sem horas de contacto)

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

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

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

- * *Conceitos de base para a análise e síntese de algoritmos distribuídos.*
- * *Abstrações fundamentais para a construção de sistemas distribuídos e a sua realização algorítmica.*
- * *Técnicas para melhorar a fiabilidade e a escalabilidade dos sistemas distribuídos.*

Aptidões e competências:

- * *Construção de algoritmos distribuídos e a sua aplicação no desenvolvimento de sistemas distribuídos.*
- * *Análise de algoritmos distribuídos.*
- * *Programação de sistemas distribuídos.*

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

- * *Basic concepts to the analysis and synthesis of distributed algorithms.*
- * *Fundamental abstractions for building distributed systems and the algorithms that are used to realize them.*
- * *Techniques for improving the reliability and scalability of distributed systems.*

Skills and competences:

- * *Designing distributed algorithms and applying them to build distributed systems.*
- * *Analysing distributed algorithms.*
- * *Programming distributed systems.*

4.4.5. Conteúdos programáticos:*1. Modelos de Computação Distribuída:*

1. *Modelação de processos, falhas, primitivas criptográficas.*
2. *Modelos temporais: Síncrono, Assíncrono and Sincronia Eventual.*
3. *Primitivas de Comunicação: (melhor-esforço, exactamente uma vez, broadcast).*
2. *Sistemas entre-pares (P2P):*
 1. *Redes sobrepostas não estruturadas.*
 2. *Protocolos epidémicos.*
 3. *Redes sobrepostas estruturadas.*
 4. *Hashing consistente e encaminhamento ao nível aplicacional.*
 5. *Casos de Estudo.*

3. Acordo:

1. *Consensus em sistemas síncronos.*
2. *Consensus em sistemas assíncronos & FLP.*
3. *Paxos e algumas variantes.*
4. *Replicação e Tolerância a Falhas:*
 1. *Especificação de sistemas replicados.*
 2. *Replicação activa e Replicação passiva.*
 2. *Consistência forte: Replicação de máquina de estados.*
 3. *Consistência fraca: Teorema de CAP, consistência eventual, consistência causal.*
 4. *Casos de Estudo.*

5. Transações Distribuídas:

1. *Commit em duas fases.*
2. *Commit em três fases.*

4.4.5. Syllabus:*1. Distributed computation models:*

1. *Modelling processes, faults, crypto primitives and communication.*
2. *Synchronous, asynchronous and partially synchronous models.*
3. *Communication Primitives (Best-effort, exactly once, Broadcast).*

2. Peer-to-Peer Systems:

1. *Unstructured Overlay networks.*
2. *Gossip Protocols.*
3. *Structured Overlay networks.*
4. *Consistent Hashing and Routing.*
5. *Case studies.*

3. Consensus:

1. *Consensus in Synchronous Systems.*
2. *Consensus in Asynchronous Systems & FLP.*
3. *Paxos and some variants.*

4. Replication and Fault Tolerance:

1. *Specifying replicated systems.*
2. *Active Replication and Passive Replication.*
2. *Strong Consistency: State Machine Replication.*
3. *Weak Consistency: CAP theorem, eventual consistency, causal consistency.*
4. *Case studies.*

5. Distributed transactions:

1. *Two-phase commit.*
2. *Three-phase commit.*

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

O conteúdo programático da cadeira aborda a modelação de sistemas distribuídos e as bases fundamentais para a compreensão de propriedades, desenho, e demonstração da correção de algoritmos distribuídos. Os estudantes aplicam estes conceitos para resolver o problema da disseminação. De seguida são introduzidos a problemas de escala com base em algoritmos entre-pares: redes sobrepostas e algoritmos de gestão de informação probabilísticos e descentralizados. De seguida é estudado o problema da localização de recursos entre-pares. O programa evolui para abordar a problemática da gestão de dados com ênfase na replicação: modelos de replicação, quorums, e Consensus em sistemas síncronos. o importante resuldto FLP, e formas de contornar esse resultado: Paxos e Multi-Paxos. Culminando na introdução da máquina de estados replicada. Finalmente aborda-se consistência e isolamento, através de casos de estudo. Terminando no problema do commit atómico e os algoritmos 2-phase e 3-phase commit.

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

The programatic content of the course starts with distributes system models and core concepts for undderstanding properties, design, and analytical correctness demonstration of distributed algorithms. Studens apply these concepts to tackle the broadcast problem. Then they are introduced to problems that derive from scale, focused on peer-to-peer algorithms: overlay networks and probabilistic and decentralized information management alggorithms, culminating in the study of the resource location problem. The programme evolves to address data management challenges with enfasis on replication: models, quorums, and consensus in synchronous systems. The key result FLP is studied, and ways to circumvent are presented: Paxos and Multi-Paxos. This culminates with the state machine repication model. Finally, questions related with consistency and isolation are studied based on case studies. The programme terminates with the atomic commit probem and the 2-phase and 3-phase commit algorithms.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas explicam-se e discutem-se os tópicos da UC. A abordagem aos problemas e à explicação dos conceitos e soluções tenta ser rigorosa, mas evitando formalismo excessivo. Nos vários tópicos são estudados exemplos de sistemas relevantes. Os estudantes são desafiados nas aulas a resolverem problemas através da construção de algoritmos de forma colaborativa (guiado pelo docente).

As aulas práticas acompanham os projetos da cadeira, fornecendo aos alunos um mapeamento entre conceitos teóricos aplicados na prática. Estas aulas permitem também a apresentação e análise de casos de estudo.

Metodologia de Avaliação:

- 2 testes (ou exame em alternativa): 45% NF
- Trabalhos para casa (indicados nas aulas teóricas e entregues presencialmente): 10% NF.
- Projeto (em três fases, realizado em grupos de 3 alunos): 45% NF (100% FREQ).

4.4.7. Teaching methodologies (including students' assessment):

In lectures the various topics are explained and discussed. The approach to this is rigorous, while avoiding an excessive formalism. A set of selected systems are studied for illustrating different techniques. Students are challenged in lectures to solve problems throught the construction of algorithms in a collaborative way (guided by the professor)

Laboratory classes follow the projects closely, giving students a mapping between concepts learned in lectures and their application in practice. Labs are used to present and analyse relevant case studies.

Evaluation Methodology:

- 2 midterms (or an exam): 45% FG.
- Individual homeworks (provided in lectures and delivered in hand): 10% FG.
- Project (in three phases, executed by groups of 3 students): 45% FG (100% FREQ).

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 sobre a análise e síntese de algoritmos distribuídos e também compreender as técnicas para melhorar a escalabilidade dos sistemas distribuídos. Tal é conseguido através da uma integração cuidada entre os conhecimentos explicados na aula teórica e a aprendizagem dos conceitos através da programação, no contexto das aulas práticas e do projeto, assim como do estudo de casos de estudo adequados. As aulas tentam despertar a atenção dos alunos aos problemas através da apresentação dos problemas e da construção guiada e colaborativa de soluções, explicando o porquê de algumas alternativas falharem. Os trabalhos de casa desafiam os alunos a desenvolver este tipo de raciocínio de forma independente, e o projeto permite aos estudantes aplicarem estes conceitos na prática com a hipótese de estudarem as suas implicações práticas.

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

The teaching methodologies aim at providing an in-depth understanding concerning the analysis and synthesis of distributed algorithms as well as promoting an understanding of the techniques for improving the scalability of distributed systems. The above goals are achieved through a careful integration of the themes presented and discussed in the lectures and the programming assignments performed by the students in the context of the lab sessions and projects, where the theoretical concepts and approaches are exercised in practice. In lectures, studens are motivated to learn fundamental concepts, by the introduction of problems and challenging them, in a guided and collaborative way, to devise algorithms that solve these problems, explain why simple approaches would miss the goals of the problem. Homeworks strive to make students ddevelop these skills in an autonomous way. The project

allows students to put in practice concepts and algorithms discussed in lectures, and to observe the practical implications of different solutions.

4.4.9. Bibliografia de consulta/existência obrigatória:

Main course reading:

- *N. Lynch. Distributed Algorithms Morgan Kauffman, 1996.*
- *C. Cachin, R. Guerraoui, L. Rodrigues "Introduction to Reliable and Secure Distributed Programming", 2nd ed, Springer, 2011.*
- *Selected research papers.*

Complementary reading:

- *H. Attiya and J. Welch. Distributed Computing: Fundamentals, Simulations, and Advanced Topics (2nd Ed.), Wiley 2004.*
- *S. Mullender (editor) Distributed Systems, Second Edition, ACM Press, Addison-Wesley, MA, 1994.*
- *A.S. Tanenbaum and M. van Steen. Distributed Systems. Principles and Paradigms. (2nd Ed.) Prentice Hall, 2007.*
- *Rodrigo Rodrigues, Peter Druschel. Peer-to-Peer Systems. Communications of the ACM. Vol. 53 N10, pp 72-82.*

Mapa IV - Aprendizagem Profunda

4.4.1.1. Designação da unidade curricular:

Aprendizagem Profunda

4.4.1.1. Title of curricular unit:

Deep Learning

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

João Leite (sem horas de contacto)

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

Ludwig Krippahl - T:28; PL:28

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

Esta unidade curricular visa dotar o estudante de competências para:

Compreender:

- *Princípios básicos de 'deep learning'*
- *Fundamentos computacionais de redes profundas*
- Algoritmos de otimização, funções de ativação, funções objetivo*
- *Diferentes arquiteturas de redes profundas e sua utilidade:*
- Densas, convolução, recorrentes, modelos geradores.*
- *Treino e regularização de redes profundas*
- *A importância das características dos dados e da criação de conjuntos de treino, validação e teste*

Ser capaz de:

- *Selecionar modelos e funções objetivo adequados para cada problema*
- *Usar bibliotecas modernas de aprendizagem profunda*
- *Implementar redes profundas, otimizar os seus hiper-parâmetros e treiná-las.*
- *Avaliar o treino dos modelos e a qualidade dos resultados.*

Conhecer:

- *Tipos de problemas resolvidos com redes profundas*
- *Arquiteturas e regularização de redes profundas*
- *Métodos de seleção de modelos e hiper-parâmetros*

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

This curricular unit will provide the student with the skills to:

Understand:

- *The foundations of deep learning*
- *Fundamentals of deep network computing*
- *Optimization algorithms, activation functions, objective functions*
- *Different deep network architectures and their usefulness:*
- *Dense, convolution, recurrent, generative models.*
- *Training and regularization of deep networks*
- *The importance of data characteristics and of training, validation and test sets*

Be able to:

- *Select appropriate models and loss functions for different problems.*
- *Use modern libraries for deep learning.*
- *Implement deep networks, optimize their hyper-parameters and train them.*
- *Evaluate the training of the models and the quality of the results.*

Know:

- *Types of problems solved with deep networks*
- *Architectures and regularization of deep networks*
- *Model selection methods and hyper-parameters*

4.4.5. Conteúdos programáticos:

1. *Introdução: fundamentos de aprendizagem profunda, transformações não-lineares e sobreajustamento.*
2. *Redes neuronais artificiais, retropropagação e redes feedforward profundas.*
3. *Implementação e treino de redes neuronais profundas*
4. *Otimização e regularização de redes feedforward. Treino, teste e validação cruzada.*
5. *Redes de convolução, teoria e prática*
6. *Aprendizagem profunda não supervisionada com autoencoders.*
7. *Aprendizagem de representações e transferência de conhecimento.*
8. *Modelos geradores.*
9. *Redes recorrentes e problemas com dados sequenciais.*
10. *Aprendizagem por reforço.*
11. *Modelos probabilísticos estruturados.*
12. *Aspectos práticos da seleção, aplicação e otimização de redes profundas.*
13. *Visualização de dados e modelos.*
14. *Problemas abertos em aprendizagem profunda.*

4.4.5. Syllabus:

1. *Introduction: fundamentals of deep learning, nonlinear transformations and overfitting.*
2. *Artificial neural networks, backpropagation. and deep feedforward networks.*
3. *Implementation and training of deep neural networks*
4. *Optimization and regularization of feedforward networks. Training, testing and cross validation.*
5. *Convolution networks, theory and practice*
6. *Unsupervised deep learning with autoencoders*
7. *Representation and transfer learning*
8. *Generative models*
9. *Recurrent networks and problems with sequential data*
10. *Reinforcement learning*
11. *Structured probabilistic models*
12. *Practical aspects of deep network selection, application and optimization*
13. *Data and model visualization.*
14. *Open problems in deep learning*

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

Os conteúdos programáticos de 1 a 4 ajudam a compreender os fundamentos de aprendizagem profunda e como redes profundas são treinadas e otimizadas, tanto em teoria como na prática. Isto também contribui para a capacidade de usar bibliotecas modernas de aprendizagem profunda.

De 5 a 11 exploram diferentes tipos de problemas e arquiteturas de modelos, contribuindo para o conhecimento dos

alunos sobre o âmbito da aprendizagem profunda e compreensão de como diferentes modelos resolvem diferentes problemas. Estes pontos são abordados de forma teórica e prática, contribuindo para o conhecimento de diferentes opções, a compreensão de como se aplicam e a capacidade de as usar para resolver problemas práticos. De 12 e 13 focam aspetos práticos de seleção e otimização de modelos, numa visão unificadora que abrange os diferentes tópicos abordados anteriormente. O 14 ajuda a compreender o progresso da aprendizagem profunda e como esta disciplina se encaixa na pesquisa científica, na IA e na sociedade.

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

Points 1 - 4 of the syllabus help understand the foundations of deep learning and how deep networks are trained and optimized, both in theory and in practice. This also contributes to the ability to use modern libraries for deep learning.

Points 5 - 11 explore different types of problems and model architectures. These contribute to the students' knowledge of the scope of deep learning and to their understanding of how different deep models suit different problems. These points are covered both in theory and practice, contributing to the knowledge of the different options, understanding how they apply and ability to use them for practical problems

Point 12 and 13 focus on practical aspects of model selection and optimization, providing a unifying view that cuts across the different topics covered previously

Finally, point 14 helps understand how deep learning is progressing and how this discipline fits in scientific research, AI and our society.

4.4.7. Metodologias de ensino (avaliação incluída):

As 56 horas de contato entre alunos e professores serão divididas em 28 horas de aulas teóricas e 28 de aulas práticas.

As aulas teóricas serão apoiadas por apontamentos fornecidos pelo professor e referências aos capítulos apropriados dos livros recomendados. Cada uma dessas aulas será dividida em aproximadamente 2/3 de exposição e 1/3 de discussão livre com os alunos.

As aulas práticas consistirão de aulas de exercícios, nas quais os alunos seguem conjuntos de exercícios fornecidos pelos tutores, bem como aulas de projetos nas quais os alunos terão apoio para projetos de avaliação concebidos pelos alunos.

As notas serão determinadas por uma componente teórica composta por dois testes ou um exame e uma componente prática na qual os alunos implementam e avaliam seus próprios modelos profundos para resolver problemas práticos. As duas componentes serão igualmente importantes para a nota final.

4.4.7. Teaching methodologies (including students' assessment):

The 56 hours of contact between students and teachers will be divided into 28 hours of lectures and 28 of practical tutorial classes.

The theoretical classes will be supported by class notes provided by the lecturer and references to the appropriate chapters of the recommended textbooks. Each of these classes will be divided into approximately 2/3 exposition and 1/3 free discussion with the students.

The tutorial classes will consist both of exercise classes where students follow sets of exercises provided by the tutors as well as project classes where the students can get help for projects of their conception.

Grades will be determined by a theoretical component consisting of two tests or one exam and a practical component in which students implement and evaluate their own deep models to solve practical problems. The two components will be equally important for the final grade.

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

A divisão de horas de contato em 28 horas de palestras e 28 de aulas práticas é consistente com o objetivo de fornecer tanto uma sólida compreensão teórica de aprendizagem profunda bem como a capacidade de resolver problemas, na prática, aplicando esse conhecimento.

O sistema de classificação também reflete a igual importância da compreensão teórica e aptidão prática, dando igual peso a ambos os aspetos do curso.

Em detalhe, as primeiras semanas do curso fornecem os fundamentos teóricos de aprendizagem profunda e explicam como redes profundas são treinadas. Concedem também a capacidade prática necessária para usar bibliotecas modernas na implementação, treino e otimização de modelos profundos.

O período intermediário do curso explorará diferentes tipos de problemas e arquiteturas de modelo para resolvê-los, além de abordar detalhes práticos, como funções de custo, otimizadores e seleção de hiper-parâmetros. Estes pontos contribuem para o conhecimento dos alunos sobre o âmbito da aprendizagem profunda, diferentes problemas e soluções e também para a compreensão de como usar diferentes modelos profundos para diferentes problemas. Como há aulas teóricas e práticas a cada semana, esses tópicos serão abordados em paralelo tanto numa perspetiva

teórica quanto prática.

A última parte do curso abrange tópicos mais amplos, que não se restringem a problemas específicos ou tipos de rede, mas que são de interesse geral em aprendizagem profunda. Esta visão unificadora de aspetos transversais da aprendizagem profunda ajuda a consolidar a compreensão dos tópicos abordados e de como usar com sucesso modelos profundos na prática.

Finalmente, o curso termina com uma amostra de problemas abertos em aprendizagem profunda, desde aspetos técnicos até questões mais abrangentes como o papel da aprendizagem profunda em IA e às implicações sociais dessas técnicas. Isso contribuirá para dar aos alunos uma compreensão de como a aprendizagem profunda se enquadra nestes contextos da sociedade, da inteligência artificial em geral e da investigação científica.

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

The division of contact hours in 28 hours of lectures and 28 of practical tutorial classes is consistent with the goal of providing both a solid theoretical understanding of deep learning as well as the ability to solve problems, in practice, by applying this knowledge.

The grading system also reflects this equal importance of theoretical understanding and practical ability, giving equal weight to both aspects of the course.

In detail, the first weeks of the course provide the theoretical foundations of deep learning and how deep networks are trained and also the practical skills necessary for using modern libraries in the implementation, training and optimization of deep models

The mid period of the course will explore different types of problems and model architectures to solve them, as well as covering practical details such as loss functions, optimizers and hyper-parameter selection. These contribute to the students' knowledge of the scope of deep learning, different problems and solutions, and also to their understanding of how to use different deep models for the different problems. Since there are lectures and tutorial classes every week, these topics will be covered both from theoretical and practical perspectives.

The latter part of the course cover broader topics that are not bound to specific problems or network types but can be of general interest in deep learning. This unifying view of cross-cutting aspects of deep learning helps to consolidate the understanding of the topics covered and how to successfully use deep models in practice.

Finally, the course ends with a sample of open problems in deep learning, ranging from technical aspects to more wide ranging questions of how deep learning connects to AI in general and the social implications of these techniques. This will contribute to give students an understanding of how deep learning fits in our society, the general problem of artificial intelligence and scientific research.

4.4.9. Bibliografia de consulta/existência obrigatória:

Main textbook, mandatory:

Goodfellow, Ian, Bengio, Yoshua, Courville, Aaron: Deep Learning , MIT Press, 2016

Complementary reading:

Skansi, Sandro: Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence , Springer, 2018

Géron, Aurélien: Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems O'Reilly Media, Inc, 2017

Mapa IV - Arquitetura e Protocolos de Redes de Computadores

4.4.1.1. Designação da unidade curricular:

Arquitetura e Protocolos de Redes de Computadores

4.4.1.1. Title of curricular unit:

Computer Networks Architecture and Protocols

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***José Augusto Legatheaux Martins - T:14; PL:14***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Paulo Orlando Reis Afonso Lopes - T:14; PL:14***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Objetivos de compreensão e conhecimento*

- *Modelos base das redes e ferramentas usadas para as estudar*
- *Requisitos e soluções de encaminhamento*
- *Diferenciação da qualidade de serviço e técnicas de implementação*
- *Tópicos emergentes em redes de computadores baseados nos requisitos e soluções das redes para centros de dados*

Objetivos de aprendizagem

- *Como montar uma rede TCP/IP*
- *Capacidade de usar ferramentas de análise e medida*
- *Contacto com os equipamentos usados em redes de computadores*
- *Utilização de emuladores de redes*
- *Utilização de clusters de servidores*
- *Investigar e analisar tópicos emergentes na literatura científica*

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

- *Basic models and tools for networking*
- *Routing requirements and solutions*
- *Differentiation of quality of service and implementation tools*
- *Emergent topics in computer networking based on the requirements and solutions for data centre networking*

Know-how goals

- *Ability to setup a small TCP/IP Network*
- *Ability to use tools for network analysis and measurement*
- *Awareness of representative equipment used in real computer networks*
- *Usage of network emulators*
- *Usage of servers clusters*
- *Research and analyze papers on emergent networking topics*

4.4.5. Conteúdos programáticos:

1. *Flooding, Spanning tress, VLANs*
2. *Shortest-path routing and link state protocols*
3. *Princípios, modelos e ferramentas das redes de computadores*
4. *endereçamento IP*
5. *BGP*
6. *Controlo da saturação na orla e no interior da rede*
7. *Ferramentas para diferenciar qualidade de serviço*
8. *Utilização de túneis e uma introdução ao MPLS*
9. *Requisitos das aplicações e padrões de tráfego nos centros de dados*
10. *Estrutura física da rede, máquinas virtuais e sua migração*
11. *Software Defined Networking (SDN)*
12. *Gestão e arquitectura das redes de centros de dados*
13. *Optimização da rede do centro de dados*
14. *TCP para centros de dados*

*Laboratório especial com equipamento convencional de encaminhamento para estudo dos protocolos STP, OSPF, etc.
Exercícios com recurso ao emulador Mininet e virtual switches.*

4.4.5. Syllabus:

1. *Flooding, Spanning tress, VLANs*
2. *Shortest-path routing and link state protocols*
3. *Networks principles, models and tools*
4. *IP addressing*
5. *BGP*

6. Congestion control at the edge and at the network level
7. Quality of service tools
8. Tunnels utilization and a short introduction to MPLS
9. Data centre application requirements and traffic patterns
10. Physical structure of the network, machine virtualization and migration
11. Software Defined Networking (SDN)
12. Data centre network architecture and management
13. Traffic engineering for intra- and inter-data centres networks
14. TCP for data centre environments

Hands-on laboratory sessions on switched Ethernet and shortest path routing take place in a special purpose laboratory equipped with conventional network gear. Programming assignments on network control and virtualization will use Mininet and Openflow virtual switches.

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

O curso tem por objetivo permitir a aprendizagem aprofundada pelos estudantes dos temas clássicos de redes de computadores dado que se trata de uma unidade curricular opcional para alunos do mestrado. Adicionalmente, dois outros objetivos são atingidos. Por um lado permitir aos alunos um contacto com switches, routers e as ferramentas usadas para teste e validação do funcionamento da rede. Este objetivo é atingido através das sessões laboratoriais. O segundo objetivo é levar os estudantes a contactarem com tópicos emergentes. Este segundo objetivo é atingido através do estudo de tópicos emergentes e de papers, assim como pela realização de um projeto de grupo geralmente centrado na temática das redes para centros de dados.

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

The course aims to expose students to the classical networking subjects to allow their deeper understanding since students take the course as an elective at senior level. Additionally, two other goals are to be fulfilled. On one hand a first contact with switches and routers as well as the tools used to test and validate network functioning and performance. This goal is mainly accomplished with the lab sessions. On the other hand students are exposed to new and emergent topics. This goal is accomplished with papers and emergent topics study as well as the realization of a group project, usually centered around data center networking.

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas têm por objectivo a apresentação dos principais temas e discutir as questões mais relevantes. Os estudantes devem, de preferência antes das aulas, estudar a bibliografia recomendada.

As sessões laboratoriais têm lugar num laboratório especializado e têm por objectivo montar diversas configurações de rede e testar o seu desempenho. O laboratório está equipado com equipamento actual, idêntico ao que é usado na indústria para montar redes de pequena e média dimensão. Estas sessões laboratoriais têm uma aproximação do tipo aprendizagem através de "mão na massa".

Algumas das aulas laboratoriais usam emuladores e outras ferramentas software.

Os projectos de grupo permitem aos estudantes ganhar competências em temas emergentes em redes de computadores.

A avaliação baseia-se em: dois testes intermédios (2 x 20%), resumo de dois artigos (2 x 5%) e dois trabalhos laboratoriais (2x25%).

4.4.7. Teaching methodologies (including students' assessment):

Lectures are intended to support the instructor's presentation of fundamental issues. Students should, preferably before classes, study the recommended bibliography.

Laboratory sessions take place in a specialized laboratory and their aim is to perform several network configurations and test their performance. The laboratory is equipped with state of the art network gear, identical to the one currently used in industry in small to medium networks. These lab sessions are based on a "hands-on" computer network learning approach.

Some laboratory classes use emulators and other software tools.

Group projects allow students to gain insights in hot and emergent computer networking topics.

Grading is as follows: two intermediate tests (2 x 20%), two article summaries (2 x 5%) and two laboratory assignments (2x25%).

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 aprofundada pelo aluno dos temas clássicos de redes de computadores dado que se trata de uma unidade curricular opcional para alunos do mestrado. Adicionalmente, dois outros objetivos são atingidos. Por um lado permitir aos alunos um contato com switches, routers e as ferramentas usadas para teste e validação do funcionamento da rede. Este objetivo é atingido através das sessões laboratoriais. O segundo objetivo é levar os estudantes a contactarem com tópicos emergentes e relevantes. Este segundo objetivo é atingido através da realização de um projeto de grupo que comporta trabalho de investigação e de implementação. O projeto está geralmente centrado em questões características dos centros de dados.

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

The course aims to expose students to the classical networking subjects to allow their deeper understanding since students take the course as an elective at senior level. Additionally, two other goals are to be fulfilled. On one hand a contact with switches and routers as well as the tools used to test and validate network functioning and performance. This goal is mainly accomplished with the lab sessions. On the other hand students are exposed to new and emergent topics. This goal is accomplished with the realization of a group project that can encompass research as well as implementation work. The project is generally related with topics of data center networking.

4.4.9. Bibliografia de consulta/existência obrigatória:

Course support books

- José Legatheaux Martins, "Fundamentos de Redes de Computadores - Ilustrados com base na Internet e nos Protocolos TCP/IP," 1ª Edição, FCT/UNL, 2018.
- Larry L. Peterson and Bruce S. Davie, "Computer Networks – A Systems Approach – 4th Edition," Morgan & Kaufman, 2007

Course complement books

- Hamed Haddadi and Olivier Bonaventure (editors), "Recent Advances in Networking", Volume 1, ACM SIGCOMM eBook, August 2013
- Ivan Marsic, "Computer Networks – Performance and Quality of Service," Rutgers University, 2013 (<http://www.ece.rutgers.edu/~marsic/books/CN/>)

Mapa IV - Computação de Alto Desempenho**4.4.1.1. Designação da unidade curricular:**

Computação de Alto Desempenho

4.4.1.1. Title of curricular unit:

High Performance Computing

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Hervé Miguel Cordeiro Paulino (Regente) - T:28; 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):

Disciplina de especialização que fornece competências na área de High Performance Computing, i.e. estudo das metodologias/técnicas que permitem explorar arquiteturas hardware com múltiplos processadores, de modo a diminuir os tempos de execução de programas com necessidade de recursos de computação elevados. O curso inclui uma vertente aplicada/experimental com atividade laboratorial.

Requisitos: conceitos básicos de arquitetura, redes e sistemas de operação.

Conhecimentos:

Arquiteturas de multiprocessadores*Técnicas de decomposição de problemas e padrões recorrentes de programação/execução paralelas**Linguagens, bibliotecas e ferramentas necessárias para as várias fases de desenvolvimento de um programa paralelo**Aptidões e competências:**Melhoria do desempenho de aplicações.**Utilização de ambientes de programação de multiprocessadores**Programação de multiprocessadores com C/C++ OpenMP, CUDA e Apache Spark***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):**

The course is included in the degree's specialization part. The main goal is to give competences in the area of High Performance Computing; i.e. to the methodologies and techniques that allow the exploitation of hardware architectures with multiple processors, in order to reduce the execution times of programs that need high computational resources. The course also takes a hands-on approach with laboratory and project activity. Students should have knowledge about computer architecture, computer networks and operating systems.

*Knowledge and understanding goals**Multiprocessor architectures**Techniques for problem decomposition**Languages, libraries and tools needed for all the development cycle of a parallel program**Know-how goals**Reduction of application execution time.**Development frameworks for symmetric multiprocessors and clusters**Multiprocessor programming using C/C++ OpenMP, CUDA and Apache Spark***4.4.5. Conteúdos programáticos:***Motivação*

- *Porquê computação paralela?*
- *Porquê computação de alto desempenho?*

Fundamentos de Computação Paralela

- *Arquiteturas Paralelas*
- *Desempenho paralelo*
- *Paradigmas de Programação Paralela*
- *Desenho de Algoritmos Paralelos*

Computação Paralela

- *Padrões e estratégias para a programação paralela*
- *Processamento em memória partilhada*
- *Computação em GPUs*
- *Programação por troca de mensagens*

Computação de alto desempenho centrada em dados

- *MapReduce*
- *Apache Spark*

Algoritmos Paralelos

- *Algoritmos para processamento de gráficos*
- *Algoritmos de aprendizagem automática*

O futuro da computação de alto desempenho

- *Desafios na indústria*
- *Tópicos de pesquisa em aberto*

4.4.5. Syllabus:*Motivation*

- *Why Parallel Computing?*
- *Why High Performance Computing?*
- *The convergence of the Big Compute and Big Data trends of thought*

Fundamentals of Parallel Computing

- *Parallel Architectures*
- *Parallel Performance*
- *Parallel Programming Paradigms*
- *Designing Parallel Algorithms*

Parallel Computing

- *Parallel Programming Patterns and Strategies*
- *Shared Memory Processing*
- *GPU Computing*

*Message-passing programming**Data-Centric High Performance Computing*

- *MapReduce*
- *Apache Spark*

Parallel Algorithms (Putting it All Together)

- *Graph processing algorithms*
- *Machine learning algorithms*

The Future of High Performance Computing

- *Challenges in the industry*
- *Open research topics*

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

O objetivo principal de fornecer competências para o uso de multiprocessadores e arquiteturas heterogêneas é atingido pela conjugação das duas partes da cadeira: a primeira em que se percorrem as metodologias e as ferramentas necessárias às diferentes fases do ciclo de vida de um programa paralelo; na segunda esses conhecimentos são exercitados e consolidados através da análise de várias áreas de aplicação em que o potencial para aplicação da computação paralela é enorme, incluindo a motivação para a necessidade do desenvolvimento de algoritmos que podem ter pouco a ver com os algoritmos sequenciais aplicados nas mesmas áreas.

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

The course aims to give competences for exploiting multiprocessors and heterogeneous architectures; this objective is achieved through the combination of the two parts of the course: in the first part, students are exposed to the methodologies and tools used in the life cycle of a parallel program; in the second part, this knowledge is consolidated by applying it to different areas where there is a great potential for parallel computing use. From this second part, contributions are made towards the idea that parallel algorithms can be very different from the sequential ones used in the same area.

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. As sessões laboratoriais têm lugar num laboratório de uso geral com acesso a PCs (que são multi-core) quer para a programação de memória partilhada quer para a de memória distribuída. Estão também acessíveis máquinas com arquiteturas heterogêneas (com CPU e GPUs Nvidia) para a programação de modelos de memória partilhada.

Componentes da avaliação:

Dois testes intermédios (com o peso total de 60% na nota final)

Dois projetos de programação (com o peso total de 40% na nota final)

4.4.7. Teaching methodologies (including students' assessment):

Lectures are intended to support the instructor's presentation of fundamental issues.

Laboratory sessions take place in a general-purpose laboratory with PC-based multi-core nodes for shared memory and distributed-memory programming. Access to machines with heterogeneous architectures (CPUs and GPUs Nvidia) for shared memory programming.

Assessment Components:

Two intermediate tests (60 % of the final grade)

Two programming assignments (40% of the final grade)

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

As duas partes da cadeira contribuem para os objetivos da aprendizagem. Para qualquer das partes, a compreensão e o estudo dos tópicos principais é objeto das aulas teóricas e avaliado através de testes.

Quanto às sessões laboratoriais, eles são de natureza diferente nas duas partes. Na primeira parte, as sessões laboratoriais consistem na resolução de problemas de pequena dimensão que permitem experimentação com bibliotecas e ferramentas que implementam os conceitos expostos. Na segunda parte, as aulas práticas são usadas para a discussão das soluções a usar nos projetos de programação; as técnicas e os conceitos expostos na 1ª parte da cadeira são revisitados e o seu uso contribui de forma fundamental para serem atingidos os objetivos de aprendizagem.

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

Both parts of the course contribute to the learning objectives. Comprehension of the main topics is dealt by lectures and evaluated with tests.

Laboratory sessions have different characteristics in the two parts. In the first part, practical classes consist of the use of libraries and tools that implement the methodologies and techniques exposed in the corresponding lectures.

In the course part related with applications and algorithms for selected areas, laboratory sessions are dedicated to discussion of particular aspects of the solution development. The concepts and techniques of the first part of the course are revisited and their use makes a fundamental contribution to the fulfilment of the learning objectives.

4.4.9. Bibliografia de consulta/existência obrigatória:

As usual in HPC courses, there is no required textbook. There are several books that cover the fundamental concepts of HPC. A list follows:

• *T Sterling, M Brodowicz, M Anderson. High Performance Computing - Modern Systems and Practices. Morgan Kaufmann, 2017.*

• *T Rauber and G Rünger. Parallel Programming for Multicore and Cluster Systems. Springer, 2013*

• *P Pacheco. An Introduction to Parallel Programming. Morgan Kaufmann, 2011*

• *Ian Foster. Designing and Building Parallel Programs. Addison-Wesley, 1995, GPU computing:*

• *NVIDIA documentation*

• *J Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley, 2010*

Data-intensive computing:

- *A Shook and D Miner. MapReduce Design Patterns. O'Reilly Media, 2012*
- *High Performance Spark. O'Reilly Media, 2017*

Mapa IV - Computação Multimédia

4.4.1.1. Designação da unidade curricular:

Computação Multimédia

4.4.1.1. Title of curricular unit:

Multimedia Computing

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Nuno Manuel Robalo Correia - T:28; 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):

Saber

- *As características principais dos diferentes tipos de informação.*
- *Algoritmos de análise e processamento de imagem, áudio e vídeo.*

Fazer

- *Escolher o formato mais apropriado para um determinado tipo de informação.*
- *Definir a estrutura subjacente a uma aplicação multimédia.*
- *Programar bibliotecas para análise, processamento e visualização.*
- *Programar algoritmos de processamento.*
- *Programar aplicações multimédia.*

Soft-Skills

- *Entender o carácter multidisciplinar da disciplina e a relação com outras áreas.*
- *Capacidade para gerir e desenvolver parcialmente um projeto multimédia.*

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

Knowledge

- *The main characteristics of the different types of information.*
- *Image, audio and video analysis and processing algorithms.*

Application

- *Choose the most appropriate format for a particular type of multimedia information.*
- *Define the architecture of a multimedia application.*
- *Program libraries for analysis, processing and visualization.*
- *Program analysis and processing algorithms.*
- *Program multimedia applications.*

Soft-Skills

- *Understand the multidisciplinary nature of the area and the relationship with other areas.*
- Ability to manage and partially develop a multimedia project.*

4.4.5. Conteúdos programáticos:

Informação estática e informação dinâmica
Texto, imagens, áudio, vídeo
Formatos e normas de representação, compressão e transmissão
Perceção e informação.
Análise, processamento e visualização de informação
Histogramas e operações sobre histogramas.
Convolução. Exemplos.
Operações morfológicas
Características globais: cor, textura, histogramas de contornos.
Características locais: matching, alinhamento de imagens, cantos, SIFT.
Deteção de faces.
Análise e processamento de áudio.
Perspetivas futuras em computação multimédia

4.4.5. Syllabus:

Static and dynamic information
Text, images, audio, video
Formats and standards of representation, compression and transmission
Perception and information.
Information analysis, processing and visualization
Histograms and operations on histograms.
Convolution. Examples.
Morphological operations
Global features: color, texture, edge histograms.
Local features: matching, image alignment, corners, SIFT.
Face detection.
Analysis and audio processing.
Perspectives on multimedia computing

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

O programa inclui o processo e os métodos para a construção de sistemas e aplicações multimédia. São abordados os diferentes tipos de informação multimédia e a sua representação computacional. São estudados e experimentados algoritmos de análise e processamento de imagem, áudio e vídeo. Estudam-se e experimentam-se bibliotecas de suporte à programação, no âmbito de aplicações que envolvem manipulação e processamento de informação. É apresentada e discutida investigação recente na área e são analisadas as perspectivas.

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

The program includes the process and methods for the development of multimedia systems and applications. The different types of multimedia information and its computational representation are addressed. Algorithms for image, audio and video analysis and processing are studied and experimented. In the context of applications that involve multimedia information processing, support libraries are studied and experimented. Recent research in the field is presented and discussed and future perspectives are analyzed.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas é apresentada a matéria, com exemplos de sistemas relacionados. O laboratório corresponde ao desenvolvimento de uma aplicação multimédia com componentes de interface e de processamento de informação.

A avaliação inclui dois testes escritos individuais e 2 componentes de projeto, que incluem algoritmos de imagem, áudio e vídeo para construção de sistemas multimédia. Os alunos que não obtiveram aprovação nos testes poderão realizar um exame, cuja nota substituirá a nota dos testes.

Testes sem consulta (50%) + Componente prática com dois momentos [1. Especificação + Estado Arte 2. Código+Relatório] (50%).

Nota mínima: 9.5 a cada uma das componentes.

4.4.7. Teaching methodologies (including students' assessment):

In the lectures the course content is presented, illustrated with application examples and references to related work. The laboratory corresponds to the development of a multimedia application with interface and information processing components.

The evaluation of the course comprises the following elements: two mid-term written individual tests and two project components that include image, audio and video algorithms for building multimedia systems. Students approved in the project and not approved in the tests may also do a final exam, and the resulting grade replaces the grades of the tests.

*Tests (50%) + Practical component with two moments
[1. Specification + State of Art 2. Code + Report] (50%).*

Minimum score: 9.5 for each component.

4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular:
Nas aulas teóricas são expostos os princípios fundamentais da unidade curricular, acompanhados de exemplos e de referências para trabalho relacionado. Nas aulas práticas os alunos desenvolvem um projeto de dimensão média, que acompanha os tópicos da componente teórica.

Para além da bibliografia recomendada, os alunos têm acesso aos slides apresentados nas aulas teóricas e a artigos científicos, de forma a enquadrar os tópicos e o trabalho no contexto mais genérico dos sistemas multimédia atuais.

Durante as aulas práticas, os alunos têm que trabalhar em grupos (preferencialmente de dois elementos) e aplicar os conhecimentos proporcionados pelas aulas teóricas. Os projetos têm uma dimensão apreciável de forma a serem semelhantes a um trabalho de equipa real, no desenvolvimento de sistemas multimédia.

Para obter aprovação, o aluno deve assistir a pelo menos dois terços das aulas práticas. Os testes escritos individuais permitem a avaliação mais precisa dos conhecimentos adquiridos por cada aluno.

4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:
The fundamental principles are presented during the lectures, illustrated with examples and references to related work. In the laboratory classes the students develop projects that implement the topics of the theoretical component.

In addition to the recommended readings, students have access to the slides presented in the lectures and related scientific, to contextualize the topics and the project in the context of current multimedia systems.

During the laboratory classes, students have to work in groups (preferably with two elements) and apply the knowledge obtained from the lectures and other education materials. The projects are relatively replicating as much as possible the situation of a real team developing a multimedia system.

To obtain approval, the student must attend at least two thirds of the practical classes. The individual written tests allow a more accurate assessment of knowledge acquired by each student.

4.4.9. Bibliografia de consulta/existência obrigatória:

- *Multimédia e Tecnologias Interactivas, Nuno Ribeiro, FCA, 2004 (2ª Edição, 2006).*
- *Tecnologias de Compressão Multimédia, Nuno Ribeiro e José Torres, FCA, 2009.*
- *Fundamentals of Multimedia, Ze-Niam Li, Mark S. Drew, Pearson, Prentice Hall, 2004.*
- *Computer Vision: Algorithms and Applications, Rick Szeliski, Springer, 2010.*
- *Fundamentals of hearing, William Yost, AP, 2013 (5ª Edição)*
- *Additional readings will be provided during classes and through the course website (including lecture slides).*

Mapa IV - Conceção e Implementação de Aplicações para a Internet

4.4.1.1. Designação da unidade curricular:

Conceção e Implementação de Aplicações para a Internet

4.4.1.1. Title of curricular unit:

Internet Application Design and Implementation

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***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:***João Ricardo Viegas da Costa Seco - T:28; PL:28***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Saber*

- 1. Aspetos fundamentais dos padrões arquiteturais e arquiteturas de software*
- 2. Princípios do desenvolvimento de progressive web applications*
- 3. Métodos de especificação e computação baseada em web services*
- 4. Estrutura interna de um browser como cliente*
- 5. Princípios do desenvolvimento centrado nos dados e no utilizador*
- 6. Mecanismos de abstração utilizados em aplicações para a Internet*
- 7. Principais ameaças à performance e como as evitar*
- 8. Mecanismos de especificação e implementação de políticas de segurança*

Fazer

- 9. Utilização de frameworks para implementar arquiteturas e estilos arquiteturais*
- 10. Especificar e construir aplicações para a Internet e para a Nuvem*
- 11. Especificar e construir clientes com comportamentos rico e reativo*
- 12. Implementar mecanismos de autenticação e especificação de regras de segurança nucleares*
- 13. Especificar e usar de forma eficiente frameworks de Object Relational Mapping*

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

- 1 - aspects of architectural patterns and software architectures for Internet Applications*
- 2 - principles of the development of progressive web applications*
- 3 - specification and implementation of web services orchestrations*
- 4 - internal structure of a browser*
- 5 - principles of data-centric and user-centric development*
- 6 - data abstraction mechanisms used in Internet applications*
- 7 - performance pitfalls of Internet applications and their workarounds*
- 8 - specification and implementation mechanisms for security policies in Internet Applications*

To Do

- 9 - use of frameworks to implement architectural styles*
- 10 - specify and build web and cloud applications*
- 11 - specify and build client applications with reactive and rich behaviour*
- 12 - implement authentication mechanisms and specify the core security rules*
- 13 - specify and efficiently use abstraction data layers such as Object Relational Mappings*

4.4.5. Conteúdos programáticos:*A - Arquitetura de software para aplicações para a Internet: Padrões Arquiteturais, Software as a Service (SaaS). Arquiteturas Web e Orientadas por Serviços. Progressive Web Applications.**B - Especificação de aplicações web e aplicações baseadas em serviços: Data-centric (baseada em recursos), user-centric (baseada em operações e fluxo de interação), modelação de processos e orquestração de serviços.**C - Abstrações de controlo e dados para aplicações web e de serviços (REST e SOAP). Object Relational Mapping. Modelos de dados relacionais e NoSQL. Linguagens de manipulação de dados. Modelos de programação reativa. Linguagens unificadoras para aplicações para a Internet. Arquiteturas para validação OO.**D - Desempenho, escalabilidade e monitorização de aplicações. Ameaças ao desempenho em ORMs. Filas de espera. Cache. Pesquisa e indexação. Logging.**E - Segurança em aplicações para a Internet: modelos de controlo de acesso, autenticação por terceiros. Ataques comuns.***4.4.5. Syllabus:***A - Software Architecture for Internet applications: Architectural patterns. Software as a service (SaaS). Web and Service oriented architectures. Progressive web applications.*

B - Specification of web and service applications: Data-centric (resource-based) application modelling. User-centric (operation based) specification. (IFML) Business process modelling in web applications (BPMN). Web service orchestration (BPEL).

C - Data and control abstractions of web and service applications: REST and SOAP. Object Relational Mappings. Relational and NoSQL data layer models. Data manipulation languages. Reactive programming models, Unified languages for Internet applications. OO based data validation architectures.

D - Performance, scalability and monitoring of applications: Pitfalls of ORMs. Load balancing. Queuing. Caching. Search and indexing. Logging and monitoring.

E - Security of internet applications: Access control models. Third-party authentication. Common attacks.

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:
Os objetivos desta unidade curricular relaciona-se com o seu programa da seguinte maneira:

Os conceitos de desenvolvimento de aplicações de larga-escala a partir de métodos de especificação e técnicas de desenho de software (objetivos 1,2,3,4) são atingidos pelos pontos A e B do programa. Os objetivos 10 e 11 são também atingidos pelos mesmos pontos do programa.

Os conceitos relacionados com o desenvolvimento centrado nos dados e no utilizador são ensinados pelo ponto B do programa (objetivo 5), que é suportado pela utilização de abstrações específicas do domínio (objetivos 4,6 e ponto C do programa). Isto tem como resultado transmitir as capacidades de um programador full-stack.

Outros objetivos transversais e de características não funcionais, objetivos 7 e 8, são cobertas pelos pontos D e E do programa.

Os objetivos de 9 a 14, resultam do trabalho hands-on, baseado em projetos, que apesar de pequena dimensão, exercitam todos os aspetos de arquiteturas complexas.

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

The goals of this course can be summarized and supported by the syllabus in the following way:

This course introduces the main concepts involved in the development of large-scale applications from specification methods to software design techniques (goals 1, 2, 3, 4), these goals are attained by points A and B of the syllabus. Goals 10 and 11 are attained by these syllabus points.

The concepts related to data-centric and user-centric development are taught in points B (goals 5), which build on the use of specific domain-related abstractions (goals 4,6 and point C). This results in the learning skills of a full-stack developer.

Other cross-cutting and non-functional concerns, stated in goals 7 and 8, are covered by points D and E of the syllabus.

The goals 9 to 14, result from working on hands-on project development, that is small in length but exercises the full complexity of the architectural challenges. The project is seeded by starter code provided as a full working project.

4.4.7. Metodologias de ensino (avaliação incluída):

Esta unidade curricular introduz os diferentes tópicos realçando boas práticas de desenvolvimento, e métodos de especificação e implementação. As metodologias e tecnologias são apresentadas de uma forma integrada para mostrar as especificidades no domínio do desenvolvimento de aplicações para a internet.

As aulas de laboratório são compostas por um crescendo de desafios de programação e de suporte ao desenvolvimento de um projeto.

*A avaliação escrita (valorizada em 50% da nota final) é composta por dois testes ou um exame
A componente prática (valorizada em 50% da nota final) consiste no desenvolvimento de um projeto de dimensão média por equipas de 3 elementos, entregue em duas fases. A avaliação é concluída com um relatório escrito e uma apresentação do projeto. Não há nota mínima em nenhuma das componentes, nem a noção de frequência.*

4.4.7. Teaching methodologies (including students' assessment):

The course lectures introduce different topics highlighting good development practices and specification and implementation methods. Methodologies and technologies are presented in an integrated way to show the specificities of the domain of internet application development.

Laboratory classes are composed of early development challenges and support to project development.

Written evaluation component (worth 50% of the final grade) comprises two tests or an exam. Laboratory work component (worth 50% of the final grade) comprises project development of an Internet Application with two phases. The evaluation is concluded with a written report and the presentation of the developed work.

There is no minimum grade on any of the above components, and there is no frequency condition that saved from one year to another.

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

As aulas teóricas desta unidade curricular discutem os conceitos relacionados com a arquitetura de software e padrões de desenho e arquiteturas específicos das aplicações para a internet. Adicionalmente, introduzem métodos para especificar e utilizar abstrações e práticas que promovem a construção de software eficaz. A especificidade das aplicações baseadas em serviços, progressive web applications, e outros tipos de clientes na internet são usados como exemplos. As estruturas usadas nesses exemplos são explicados à luz dos tópicos a serem expostos nas aulas teóricas.

As aulas teóricas são acompanhadas por aulas de laboratório que desafiam os estudantes a montar arquiteturas para aplicações baseadas em serviços no lado do servidor e a utilização de componentes e linguagens declarativas para construção da interface de utilizador. Um dos fatores que se impõe também é a utilização de ferramentas e plataformas de desenvolvimento de nível industrial. Utilizam-se ferramentas de especificação formal de interfaces e ferramentas para gerar código que está sempre de acordo com a especificação. Outras práticas de desenvolvimento como a utilização de controlo de versões e desenvolvimento guiado por testes colaboram para uma aprendizagem sólida de práticas de engenharia.

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

This course lectures discuss the concepts of software architecture and software patterns specific to Internet Applications. It also introduces methods to specify and use abstractions and practices that promote effective software construction practices. The specificities of service-based applications, progressive web applications, and other Internet clients are highlighted by examples and used structures are explained.

The course lectures are in tandem with the lab classes that challenge students to assemble software architectures for service-based applications on the back-end and declarative and compositional interface components on the front-end. The use of industrial-grade development frameworks, formal tools to specify interfaces and tools to generate documentation and code also contributes to solid full-stack learning. Other basic development methodologies like version control and test-driven development are followed and add to the teaching of engineering practices.

4.4.9. Bibliografia de consulta/existência obrigatória:

Brambilla, M., & Fraternali, P. (2015). Interaction Flow Modeling Language. Interaction Flow Modeling Language. <http://doi.org/10.1016/B978-0-12-800108-0.00010-2>
Fowler, M. (2002). Patterns of Enterprise Application Architecture. Source (Vol. 48). <http://doi.org/10.1119/1.1969597>
Abbott, M. L., & Fisher, M. T. (2009). The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise.
Scarioni, C., Deinum, M., Rubio, D., Josh, L., Mak, G., Wheeler, W., ... Penchikala, S. (2011). Spring IN Action 3rd. North. <http://doi.org/10.1007/978-1-4302-3346-6>
Spring.io official guides and tutorials
ReactJS official tutorials

Mapa IV - Confiabilidade de Sistemas Distribuídos

4.4.1.1. Designação da unidade curricular:

Confiabilidade de Sistemas Distribuídos

4.4.1.1. Title of curricular unit:

Dependable 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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Nuno Manuel Ribeiro Preguiça (sem horas de contacto)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Henrique João Lopes Domingos - T:28; PL:28***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Saber*

- *Conceitos de base para a análise e síntese de mecanismos e serviços de confiabilidade para distribuídos, tendo em vista a sua modelação, conceção e especificação*
- *Abstrações fundamentais para a construção dos mecanismos, técnicas e algoritmos para sistemas distribuídos confiáveis e sua realização*
- *Técnicas para combinação de propriedades de segurança, privacidade, fiabilidade, tolerância a falhas e tolerância a intrusões, para conceção de sistema distribuídos confiáveis*

Fazer

- *Construção de mecanismos e serviços para suporte de sistemas distribuídos confiáveis para suporte de aplicações e sistemas críticos*
- *Análise e avaliação experimental de mecanismos e serviços de confiabilidade num sistema distribuído para grande escala*
- *Programação e desenvolvimento prático de mecanismos de suporte a propriedades de confiabilidade*

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

- *Concepts, principles, paradigms to the analysis and synthesis of dependable distributed systems, for modeling, design goals and specification*
- *Foundations and abstractions for the design and construction of mechanisms and services for dependable distributed systems*
- *Techniques to combine security, privacy, reliability, fault-tolerance and intrusion tolerance for improved dependability properties by design*

Application

- *Designing mechanisms and services, including their components and algorithms to build critical distributed systems and applications*
- *Analysis and experimental assessment of dependable properties in dependable distributed systems*
- *Programming and development of mechanisms for dependable distributed systems in large-scale settings*

4.4.5. Conteúdos programáticos:*Introdução**Modelos de confiabilidade para sistemas distribuídos: propriedades, critérios de concepção e tipologias de suporte**Canais de comunicação segura e confiável**Prevenção, deteção, recuperação de intrusões**Replicação e Consensus em sistemas assíncronos com tolerância a falhas**Mecanismos e serviços para tolerância a intrusões**Plataformas Blockchain e modelo de gestão de confiança descentralizada; tipos de plataformas, planos de serviços, mecanismos de suporte e ambientes de programação**Gestão confiável de dados com preservação de privacidade: gestão confiável de dados privados em bases de dados e outras soluções de armazenamento em infraestruturas de entidades externas e em soluções e serviços em nuvens de computação e armazenamento de dados**Programação de software com plataformas de computação confiável para sistemas de computação móvel ou plataformas IoT e para serviços em plataformas de nuvens de computação e gestão de dados***4.4.5. Syllabus:***Introduction**Dependability models for large-scale distributed systems: characteristics, properties, design principles and support typology**Secure and reliable communication channels**Intrusion detection, intrusion prevention and intrusion recovery**Replication techniques and Consensus for asynchronous and fault-tolerant distributed systems**Mechanisms and services for intrusion tolerance guarantees**Blockchain platforms and decentralized ledgers: platform types, service planes, integrated mechanisms and programming environments*

Privacy-preserving data management for dependable distributed systems: techniques and design approaches with databases, outsourced computing and storage infrastructures and cloud services
Trustworthy software and trusted computing platforms for mobile and IoT platforms, and for cloud-enabled applications and services

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

A unidade curricular de Confiabilidade de Sistemas Distribuídos (CSD) visa fornecer conhecimentos e competências para conceção, desenvolvimento e avaliação experimental de sistemas distribuídos confiáveis para operação em ambientes de grande escala. O enfoque visa dotar os alunos de competências para lidarem com a complexidade desses sistemas e os requisitos de propriedades combinadas de segurança, privacidade, alta-disponibilidade e fiabilidade, tendo em vista a operação com tolerância a falhas ou tolerância a intrusões, bem como o suporte de mecanismos de prevenção, deteção e recuperação dessas falhas.

Na componente prática, os conceitos estudados na parte teórica são colocados em prática desenvolvendo mecanismos e serviços de confiabilidade em projeto de sistemas distribuídos dotados de propriedades de confiabilidade, dotando os alunos de práticas “hands-on” sobre ferramentas e técnicas de desenvolvimento deste tipo de sistemas.

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

The CSD course aims to provide students with specialized knowledge bases and know-how skills to develop dependable distributed systems with required complexity and support of different dependability criteria including security, privacy, reliability, high-availability properties in the context of fault tolerance and intrusion tolerance guarantees. To teach this goal, the course starts by presenting the main problems and challenges of dependability concerns in distributed systems, as well as, models and architectures targeted in designing such systems.

In the labs, the topics studied in the lectures are exercised with tools and programming techniques for the development of trusted and dependable mechanisms and services for distributed systems with increasing complexity, allowing students a hands-on approach to improve the experience in programming dependable distributed systems, particularly targeted for large-scale operation settings.

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas e materiais da disciplina são preparados em língua inglesa.

A unidade curricular é composta por aulas teóricas, onde são transmitidos e discutidos os conceitos fundamentais e aulas práticas, desenvolvidas em ambiente laboratorial.

Nas aulas práticas são feitas apresentações de técnicas de segurança, podendo envolver demonstrações dessas técnicas (de acordo com o programa das aulas práticas) e realização de exercícios básicos relacionados com essas técnicas. Algumas aulas práticas serão dedicadas a suportar os dois projetos a realizar pelos alunos, incluindo o esclarecimento de dúvidas, apresentação ou demonstração de resultados bem como discussão de possíveis soluções.

Avaliação: dois testes (25% cada), dois mini-projetos (25% cada).

4.4.7. Teaching methodologies (including students' assessment):

The classes and materials are prepared for the course to be taught in English.

The course is organized in lectures for presenting and discussing foundations, concepts, principles, paradigms, techniques or algorithms, covering the course program topics, as well as, to conduct specific discussions, analysis and clarifications on suggested readings.

Labs are organized for presenting security techniques, involving the demonstrations of these techniques and the execution of small exercises related with these techniques. Some labs are planned to support students in the development of the two mini-project assignments, including discussion and clarification of requirements and design criteria, and orientation on implementation options.

Assesment: two tests (25% each), two mini-projects (25% each).

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

A U, como disciplina de especialização, transmite conhecimentos e aptidões para concepção e desenvolvimento de sistemas distribuídos de grande escala, com requisitos de segurança, privacidade, fiabilidade e alta disponibilidade, tendo em vista o suporte de aplicações e serviços em sistemas críticos. No programa da disciplina estudam-se mecanismos associados a técnicas avançadas e propostas da investigação recente para tolerância a falhas e tolerância a intrusões, novas técnicas emergentes da criptografia computacional aplicada a soluções de privacidade de dados e computações, envolvendo ainda o estudo de soluções para computação confiável com atestação de software e ambientes de execução confiável, incluindo mecanismos de proteção ao nível do hardware.

As aulas teóricas transmitem fundamentos teóricos sobre mecanismos específicos, modelos, métodos fundamentos e técnicas de concepção para implementação e verificação das anteriores propriedades de confiabilidade. O programa da disciplina abarca.

Nas aulas práticas colocam-se em prática as técnicas de implementação e ensaio experimental, com exercícios de programação, utilização de ferramentas especializadas e realização de trabalhos ou mini-projetos de avaliação com cariz de especialização e com média complexidade.

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

In this specialization course, students will learn the fundamental principles for the design, modeling, implementation and evaluation of large scale dependable distributed systems. In the course approach the combination of required properties for such systems, including security, privacy, reliability and availability, are addressed as integrated dependability criteria for critical systems, designed and supported with fault and intrusion tolerance mechanisms, tools, techniques and algorithms.

Lectures explore the theoretical foundations, covering the different program topics.

Laboratories are organized with a strong approach on "learning by doing" or "hands-on-approach", with students conducting programming exercises with the domain of development/deployment tools and techniques, and a strong component in analytical and experimental evaluation. Through those labs and mini-projects, students can enhance their understanding of the theoretical principles, and be able to apply the acquired knowledge to address real problems.

4.4.9. Bibliografia de consulta/existência obrigatória:

Bibliography

- *W. Zhao, Building Dependable Distributed Systems, Wiley, 2014*
- *W. Stallings, L. Brown, Computer Security - Principles and Practice, Prent. Hall, 2014*
- *C. Cachin, R. Guerraoui, L. Rodrigues, Introduction to Reliable and Secure Distributed Programming (2nd Ed), Springer, 2011.*^[SEP]

Complementary Readings

- *W. Stallings, Network Security Essentials, 6th Ed. Pearson, 2017*
- *K. Birman, Reliable Distributed Computing, Springer 2005*
- *A.S. Tanenbaum and M. Van Steen. Distributed Systems Principles and Paradigms, Prent. Hall,*^[SEP] *2007*
- *M. Correia, P. Sousa, Segurança no Software, FCA Ed. 2010*

Obs) Suggested readings and selected papers from relevant research venues will be presented and proposed on the lectures. Additional materials for practical/lab activities and practical work-assignments will be available as lab materials.

Mapa IV - Desenho de Algoritmos para Problemas de Otimização

4.4.1.1. Designação da unidade curricular:

Desenho de Algoritmos para Problemas de Otimização

4.4.1.1. Title of curricular unit:

Design of Algorithms for Optimization Problems

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Pedro Manuel Corrêa Calvente Barahona - T:14; PL:14

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

Margarida Paula Neves Mamede - T:14; PL:14

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

Saber:

Definir e identificar algoritmos de aproximação e de pesquisa local.

Conhecer alguns métodos gerais para conceber algoritmos de aproximação e de pesquisa local.

Compreender heurísticas e meta-heurísticas.

Saber Fazer:

Conceber um algoritmo de aproximação e um algoritmo de pesquisa local para um problema de otimização concreto.

Calcular o rácio de aproximação de um algoritmo de aproximação.

Efetuar a análise probabilística de um algoritmo aleatório de aproximação.

Selecionar as técnicas apropriadas a um problema.

Competências Complementares:

Capacidade de investigação e autonomia.

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

Knowledge:

Define and identify approximation and local search algorithms.

Know some general methods for designing approximation and local search algorithms.

Understand heuristics and meta-heuristics.

Application:

Design an approximation algorithm and a local search algorithm for a real-world optimization problem.

Calculate the approximation ratio of an approximation algorithm.

Perform the probabilistic analysis of a randomized approximation algorithm.

Select the suitable techniques to solve a problem.

Soft-skills:

Skills in research and autonomy.

4.4.5. Conteúdos programáticos:

1. Problemas de decisão e problemas de otimização. Problemas tratáveis e problemas difíceis. Técnicas de desenho de algoritmos para problemas de otimização difíceis.

2. Algoritmos de aproximação. Rácio de aproximação. Algoritmos "greedy". Esquemas de aproximação. Programação linear e arredondamento. O método primal-dual. Algoritmos aleatórios de aproximação. Análise probabilística.

3. Algoritmos de pesquisa local. Recomeços. Pesquisa tabu. Arrefecimento simulado. Pesquisa com vizinhança variável. Colónias de formigas. Pesquisa Evolucionária Híbrida. Pesquisa Local Guiada. Pesquisa em grandes vizinhanças.

4.4.5. Syllabus:

1. Decision problems and optimization problems. Tractable problems and hard problems. Algorithm design techniques for hard optimization problems.

2. Approximation algorithms. Approximation ratio. Greedy strategies. Approximation schemes. Linear programming and rounding. The primal-dual method. Randomized approximation algorithms. Probabilistic analysis.

3. Local search algorithms. Restarts. Taboo search. Simulated annealing. Variable neighbourhood search. Ant colony optimization. Hybrid Evolutionary Search. Guided Local Search. Large neighbourhood search.

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

Os objetivos da unidade curricular podem ser sintetizados da seguinte forma:

(a) Compreender heurísticas e meta-heurísticas: Alguns algoritmos desenvolvidos no tópico 2 dos conteúdos programáticos usam heurísticas (estratégias greedy) e no tópico 3 estudam-se várias meta-heurísticas.

(b) Conceber e analisar algoritmos de aproximação e de pesquisa local para problemas concretos: Os tópicos 2 e 3 cobrem esta matéria com muitos exemplos, que ilustram e permitem aplicar os diferentes métodos estudados. O rácio de aproximação e a análise probabilística são necessários para se poderem analisar (avaliar) todos os algoritmos desenvolvidos.

(c) Selecionar as técnicas apropriadas a um problema: Os pontos fortes e fracos de cada técnica e método são discutidos ao longo dos tópicos 2 e 3.

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) Understand heuristics and meta-heuristics: Some algorithms developed in topic 2 use heuristics (greedy strategies) and several meta-heuristics are studied in topic 3.

(b) Design and analyse approximate and local-search algorithms for real-world problems: Syllabus topics 2 and 3 cover this material with many examples, which illustrate and allow all studied methods to be applied. The approximation ratio and the probabilistic analysis are needed to be able to analyse (evaluate) all developed algorithms.

(c) Select the suitable techniques to solve a problem: The strong and the weak points of each technique and method are discussed throughout topics 2 and 3.

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 um trabalho e dois testes.

O trabalho é realizado em duas fases, entregues em datas distintas, e consiste num estudo comparativo, teórico e experimental, de soluções alternativas para um problema de otimização, na elaboração de dois relatórios (um por fase) e na realização de uma discussão.

Em caso de reprovação, se a nota do trabalho não for inferior a 9, os alunos podem realizar um exame final. Os testes e o exame são com consulta.

Condição para obter aprovação: $NotaP \geq 9$ e $NotaT \geq 9$ e $NotaF = (NotaP + NotaT) / 2 \geq 10$, onde: $NotaP$ é a nota do trabalho, calculada com as notas das duas fases; $NotaT$ é a média das notas dos testes ou a nota do exame; $NotaF$ é a nota final.

4.4.7. Teaching methodologies (including students' assessment):

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

Assessment comprises one programming project and two tests.

The programming project is developed in two phases, whose deadlines are distinct, and consists in a theoretical and experimental comparative study of alternative solutions for an optimization problem, in the writing of two reports (one on each phase) and in a discussion.

In case of failure, if the project grade is not less than 9 (out of 20), students can do a final exam. The tests and the exam are open-book.

Condition to succeed: $Pgrade \geq 9$ and $Tgrade \geq 9$ and $Fgrade = (Pgrade + Tgrade) / 2 \geq 10$, where: $Pgrade$ is the project grade, calculated with the two phase 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:

Nas aulas práticas, os alunos analisam e resolvem problemas concretos, utilizando as técnicas e os métodos dados nas aulas teóricas. Os alunos começam por classificar o problema, identificando a sua classe de complexidade e o seu tipo. Depois, desenham o algoritmo, aplicando a técnica e o método sugeridos, analisam-no e implementam-no. Por fim, efetuam alguns testes experimentais. Todas as escolhas têm de ser justificadas.

Nas componentes de avaliação teóricas (testes e exame), os alunos são confrontados com questões e problemas que permitem avaliar a compreensão dos conceitos envolvidos na unidade curricular, a capacidade para efetuar escolhas apropriadas e a aptidão para desenhar e analisar algoritmos.

No trabalho, os alunos escolhem o problema que pretendem resolver e efetuam um estudo comparativo, teórico e experimental, de soluções alternativas para esse problema, utilizando técnicas e métodos lecionados. O relatório apresenta e classifica o problema, identifica os vários modelos teóricos adotados e descreve a sua aplicação, inclui a análise e a implementação dos respetivos algoritmos e contém uma análise dos resultados obtidos experimentalmente. Ao realizar o trabalho, os alunos desenvolvem a aptidão para selecionar e aplicar as técnicas apropriadas, a aptidão para conceber e avaliar soluções e a capacidade de comunicação escrita. Com este trabalho, também se pretende incentivar a autonomia e desenvolver a capacidade de investigação dos alunos.

Como em qualquer processo de aprendizagem, é fundamental que os alunos recebam feedback sobre o trabalho que vão realizando. Por esse motivo, os docentes identificam os pontos fortes e fracos de cada resolução, no final das aulas práticas ou após o trabalho ter sido avaliado.

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

In lab sessions, students analyse and solve concrete problems, making use of the techniques and methods presented in the lectures. Students start by classifying the problem, identifying its complexity class and its type. Then, they design the algorithm, applying the suggested technique and method, and perform its analysis and its implementation. Finally, they carry out some experimental tests. All choices have to be justified.

The questions and problems posed in tests and exams allow the assessment of (1) the understanding of the concepts involved in the unit, (2) the ability to make suitable choices, and (3) the ability to design and analyse algorithms.

In the programming project, students choose the problem they want to solve and perform a theoretical and experimental comparative study of alternative solutions for that problem, applying techniques and methods studied. The report presents and classifies the problem, identifies the theoretical models adopted, describes their application, includes the analysis and the implementation of the corresponding algorithms, and contains an analysis of the experimental results. By doing the programming project, students develop the ability to select and apply the appropriate techniques, the ability to create and evaluate solutions, and skills in writing communication. The project also aims at improving autonomy and research skills.

As in any learning process, students have to receive feedback on their work. For this reason, teachers identify the strong and the weak points of each solution, at the end of lab sessions or after the evaluation of the programming project.

4.4.9. Bibliografia de consulta/existência obrigatória:

Main References:

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

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

Pascal van Hentenryck and Laurent Michel. Constraint-Based Local Search. MIT Press, 2005.

Complementary References:

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

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

Mapa IV - Engenharia Orientada a Modelos

4.4.1.1. Designação da unidade curricular:

Engenharia Orientada a Modelos

4.4.1.1. Title of curricular unit:

Model-Driven 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:

168

4.4.1.5. Horas de contacto:

T:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Vasco Amaral - T:28; 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):

Compreender:

• *Vantagens e desvantagens da engenharia orientada a modelos (MDE)*

• *Riscos e oportunidades de DS(M)L vs GP(M)L*

• *Crítérios de seleção de linguagens de modelação*

Ser capaz:

• *Criar um “roadmap” de tecnologias MDE para automatização dos seus processos*

• *Transformação de modelos Modelo-para-texto e Modelo-para-Modelo*

• *Acompanhar o processo completo de Engenharia de Linguagens*

• *Usar abordagens MDE para desenvolvimento de linguagens*

• *Usar ferramentas formais e práticas (“workbenches”) de desenvolvimento de DS(M)Ls gráficas e textuais*

• *Dado um problema num domínio específico, saber como desenvolver uma DS(M)L de raiz*

• *Evoluir uma DS(M)L usando técnicas MDE*

• *Saber desenhar e executar um estudo para avaliar uma linguagem*

Conhecer:

• *A utilidade de alguns formalismos para Engenharia de Sistemas (para modelar sistemas com/sem tempo e discretos ou contínuos)*

• *Técnicas de avaliação de qualidade de DS(M)Ls do ponto de vista de usabilidade*

• *Conhecer ex. de domínio de aplicação de DS(M)Ls*

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

Understand

• *Risks and opportunities of developing and using a DS(M)L over GP(M)L*

• *Advantages and Disadvantages of MDE*

• *Criteria to choose a given modelling language for a specific modelling task*

Be able to:

- *Use formal and practical DSL (graphical and textual) development workbenches*
- *Given a problem in a specific domain, knows how to develop a DS(M) L from scratch*
- *Evolve a DS(M)L using MDE evolution techniques*
- *Know how to design and execute a study to evaluate the usability of a given language*
- *Create a roadmap of MDE technologies for automating processes*
- *How to transform Model-to-Text and Model-to-Model*
- *Follow the complete process of Language Engineering*
- *Use Model-Driven Approaches for Language Development*

Know:

- *The utility of some formalisms for System Engineering (to model a system considering or not time, discrete or continuous)*
- *Existing techniques and limitations for evaluating DS(M)Ls from the point of view of usability*
- *Some application domains of DS(M)Ls*

4.4.5. Conteúdos programáticos:

1. *Causas de Complexidade no Desenho de Sistemas*
2. *Abordagem Orientada a Modelos*
3. *Engenharia de Domínio*
4. *DS(M)Ls e Desenho de Linguagens de Modelação*
 - a) *Sintaxe Abstrata e Metamodelos (MOF, ECORE)*
 - b) *Sintaxe Concreta (Visual e Textual) e Semântica*
 - c) *Restrições invariantes (OCL,EVL)*
 - d) *Transformações de modelos*
 - *Breve introdução a alguns formalismos de modelação:*
 - a) *Causal Block Diagrams (CBDs): discretos e contínuos*
 - b) *Petri Nets*
 - c) *Statecharts*
 - d) *Event-Scheduling Discrete-Event*
 - e) *Discrete-Event System Specification (DEVS)*
5. *Verificação de modelos em MDE*
6. *Avaliação de Qualidade - Usabilidade*
7. *Breve visão geral do MDE na Engenharia de Sistemas:*
 - *Sistemas Estático e Dinâmico*
 - *Variação de tempo e sistemas dinâmicos invariantes no tempo*
 - *Formalismos de Evento Discretos e Contínuos*
 - *DEVS e equações diferenciais*
 - *AADL (linguagem de descrição da Avionics Architecture) e SysML*

4.4.5. Syllabus:

1. *Causes for Complexity in Systems Design*
2. *The Model-Driven Development approach*
3. *Domain Engineering*
4. *DSLs and Language Design*
 - a) *Abstract Syntax and Language Metamodeling (MOF, ECORE)*
 - b) *Concrete Syntax (Visual and Textual) and Semantics*
 - c) *Specifying Invariant constraints (well-formedness rules) (OCL,EVL)*
 - e) *Model-Transformations*
 - *A brief introduction to some Modelling Formalisms*
 - a) *Causal Block Diagrams (CBDs): discrete-time and continuous-time*
 - b) *Petri Nets*
 - c) *Statecharts*
 - d) *Event-Scheduling Discrete-Event*
 - e) *Discrete-Event System Specification (DEVS)*
5. *Model Checking in MDE*
6. *DSL Quality assessment - Usability*
7. *Brief overview of MDE in System Engineering:*
 - *Static and Dynamic Systems*
 - *Time Varying and Time invariant Dynamic Systems*
 - *Discrete Event formalisms*
 - *Continuous formalisms*
 - *DEVS and Differential equation specified system*
 - *AADL (Avionics Architecture Description Language) and SysML*

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

O MDE é uma abordagem padrão para abordar a complexidade do desenvolvimento de sistemas (como os CPSs) que por razões éticas, económicas ou pragmáticas têm de ser modeladas e analisadas à priori. Esta promove o uso de modelos em várias notações e suas traduções como cidadãos de primeira classe. Usam-se DSLs para poder expressar modelos em notações rigorosas adequadas que refletem as diferentes perspectivas dos sistemas, com um nível adequado de abstração utilizável pelos diferentes modeladores (potencialmente não especialistas em software). Este curso usará técnicas de MDE bem estabelecidas para modelar o domínio, projetar e desenvolver DSLs e usá-las para as diversas finalidades. Com o projeto desenvolvido durante o curso, os alunos terão a oportunidade de conceber o roteiro de modelação com diferentes camadas de abstração, começando com a definição da DSL, e através

de transformações de modelos, criarão um processo de geração automática para terminar em código, simulação ou análise.

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

MDE is a standard approach to address the complexity of systems' development (like CPS or other transdisciplinary ones). It promotes the use of models in multiple notations and their translations as first class citizens. DS(M)Ls are used to be able to express models in adequate rigorous notations that reflect the different perspectives of the systems, with an adequate level of abstraction usable by the different modelers (potentially non- software experts).

This course will use well established MDE techniques to model the domain, design and develop DS(M)Ls and to use them for several purposes. Thanks to the project developed during the course, the students will have the opportunity to conceive the modeling roadmap with different layers of abstraction, starting with the definition of the DS(M)L, and via model transformations will construct the automatic generation process to end in code or some formalisms briefly introduced in the lectures for the purpose of simulation or analysis.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas lecionam-se os conceitos teóricos e nas aulas de laboratório, que decorrerão com recurso a ferramentas.

No início das aulas os alunos receberão o planeamento para os tópicos do programa e a especificação do trabalho prático. Um prazo é dado para a entrega dos trabalhos.

As aulas teóricas são dadas em sala de aula equipada com um projetor e slides em Powerpoint. As aulas práticas são realizadas em salas equipadas com computadores, quadro e projetor.

A nota é uma média ponderada do trabalho prático de grupo (Fase I = 30%; Fase II = 30%), e das notas dos testes (Teste 1 = 20%, Teste 2 = 20%). Para obter frequência, a média ponderada dos dois testes deve ser superior ou igual a 9,5.

4.4.7. Teaching methodologies (including students' assessment):

In theoretical classes the fundamental concepts are taught and put into practice in the laboratory classes, which will be carried out using tools and examples.

At the beginning of the class the students will receive the planning for the topics of the program and the specification of the practical work. A deadline is given for the delivery of the work.

Theoretical classes are given in a classroom equipped with a projector and Powerpoint slides. The practical classes are held in rooms equipped with computers, picture and projector.

The evaluation consists of the following mandatory components: a practical work delivered in two phases; two tests. The work is carried out in a group of two students and the tests individually. Each group has a different project.

We will invite external experts (eg Quidgest, Outsystems, ...) to give a talk about their experience with the adoption of Model-Driven Engineering in their company's internal processes.

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

Nas aulas teóricas todos os tópicos do programa são apresentados, sempre ilustrados com exemplos inspirados em situações reais. Nas aulas práticas os conceitos introduzidos na teórica são consolidados com exercícios específicos e uso de ferramentas adequadas. Nas práticas também é realizado o acompanhamento dos trabalhos a fim de avaliar o seu progresso.

O trabalho prático serve para os alunos praticarem os conceitos adquiridos de forma integrada, aplicada a um caso de estudo proposto diferente de grupo para grupo. O trabalho prático fomenta o trabalho em equipa, enquanto que os testes/exames avaliam a performance individual.

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

In the lectures all topics of the program are presented, always illustrated with examples inspired by real situations. In practical classes the concepts introduced in the theory are consolidated with specific exercises. In practical classes it is also carried out monitoring of the practical work and essays to assess their progress.

The practical work is for students to practice the concepts acquired in an integrated way. The project (different for each group) fosters teamwork, while the tests/exams assess individual performance.

4.4.9. Bibliografia de consulta/existência obrigatória:

Marco Brambilla, Jordi Cabot, Manuel Wimmer, "Model-Driven Software Engineering in Practice", Morgan & Claypool Publishers, 2nd Edition, 2017

Markus Voelter, DSL Engineering: Designing, Implementing and Using Domain-Specific Languages, CreateSpace Independent Publishing Platform, 2013

Dimitris Kolovos, Louis Rose, Antonio García-Domínguez, Richard Paige, “The Epsilon Book”, The Eclipse Foundation, 2014

Bernard Zeigler, Herbert Praehofer, Tag Kim, “Theory of Modeling and simulation”, 2nd Edition, Academic Press

Mapa IV - Jogos e Simulação

4.4.1.1. Designação da unidade curricular:

Jogos e Simulação

4.4.1.1. Title of curricular unit:

Games and Simulation

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Nuno Manuel Robalo Correia (sem horas de contacto)

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

Fernando Pedro Reino da Silva Birra - T:14; PL:14

Sofia Carmen Faria Maia Cavaco - T:14; PL:14

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

Saber

- *Modelação Básica 3D e constituição de um “pipeline de visualização 3D”*
- *Técnicas e modelos de iluminação, locais e globais*
- *Arquitetura dos GPUs modernos e programação com shaders.*
- *Exploração da arquitetura dos GPUs para a produção de efeitos avançados em tempo real*
- *Técnicas específicas para o desenvolvimento de jogos e simulação.*
- *Técnicas de animação.*
- *Arquitetura de motores de jogos e sua programação.*
- *Produção e reprodução de áudio 3D*

Fazer

- *Programar parte das técnicas lecionadas.*
- *Construir aplicações gráficas 3D interativas.*
- *Programar e utilizar game engines*
- *Escolher e aplicar a técnica mais adequada à resolução de um problema.*

Soft-Skills

- *Analisar hardware gráfico e sistemas atuais*
- *Analisar e programar motores de jogos*
- *Compreensão de áreas relacionadas, como a fotografia, o cinema e a física.*

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

Knowledge

- *Basic 3D Modeling and implementation of the 3D graphics pipeline*
- *Illumination models and techniques (local and global)*
- *Modern GPU architecture and shader programming*
- *Explore GPU capabilities to create advanced effects in real time*
- *Specific techniques for game development and simulation*
- *Animation techniques*
- *Game engine architecture and development*
- *3D audio content creation and reproduction*

Application

- *Program the presented techniques*
- *Build 3D interactive graphics applications*
- *Program and use game engines*
- *Select and apply the appropriate solution to a given problem*

Soft-Skills

- *Ability to analyze current graphics hardware and current graphics systems*
- *Ability to analyze and program game engines*
- *Better understanding of related concepts such as photography, cinema and physics.*

4.4.5. Conteúdos programáticos:

1. *Hardware gráfico 3D e o pipeline gráfico 3D.*
2. *Modelação 3D e representações de geometria.*
3. *Interação da luz com os objetos, Modelos de iluminação e técnicas de sombreado.*
4. *Tópicos avançados de síntese de imagem: aplicações do uso de texturas e geração de sombras.*
5. *Grafos de cena, gestão dos dados e otimizações para tempo real.*
6. *Arquitetura de motores de jogos.*
7. *Técnicas de animação: interpolação de quadros, morphing e modelos articulados.*
8. *Deteção e resposta a colisões.*
9. *Princípios físicos dos jogos 3D e métodos numéricos.*
10. *Física da propagação do som, localização do som e síntese de som 3D.*
11. *Programação com um motor de jogos e desenvolvimento de jogos.*

4.4.5. Syllabus:

1. *3D Graphics Hardware and the 3D Graphics pipeline*
2. *Basic 3D Modeling and Geometry Representations*
3. *Light-Object interaction. Illumination models (local and global) and shading*
4. *Advanced rendering topics: texture mapping applications and shadow generation.*
5. *Game engine architecture.*
6. *Scene graph, data management and rendering optimizations for real time.*
7. *Animation techniques: key frame, vertex morphing and articulated motion.*
8. *Collision Detection and Response.*
9. *Physics for 3D Games and numerical methods.*
10. *Physics of Sound propagation, sound localization and 3D sound synthesis.*
11. *Game engine programming and game development.*

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

O programa inclui os conceitos fundamentais (1, 2, 3), técnicas e algoritmos (4, 6), para a produção de imagens de síntese em tempo real no contexto concreto do desenvolvimento de jogos 3D (5). O programa cobre ainda as técnicas principais de animação (7), simulação de fenómenos baseados em física (9) e tratamento de colisões (8), peças fundamentais na constituição dum motor de jogos 3D. A componente de áudio, com especial ênfase na sua localização, e no tratamento necessário para que essa localização seja perceptível, são também focados (10). Os alunos têm a oportunidade de aplicar de forma variada os conceitos teóricos apresentados (11) e melhorar as suas capacidades de análise no que diz respeito aos sistemas gráficos e ferramentas relacionadas.

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

The syllabus includes the fundamental aspects (1,2,3), techniques and algorithms (4,6), for real time rendering in the context of 3D game programming. The topics addressed also cover the principal animation techniques (7), physically based simulation (9) and collision detection and response (8), which are fundamental in gaming applications. The audio, with special emphasis on 3D audio synthesis is also addressed (10). Students have the opportunity to try different ways to implement the concepts presented in lectures (11) and improve their ability to criticize and analyse graphics systems and tools related with game development.

4.4.7. Metodologias de ensino (avaliação incluída):

As matérias são apresentadas dum ponto de vista formal e com ênfase na motivação para o seu estudo. A exposição é acompanhada de demonstrações práticas. Os tópicos abordados serão objeto de aplicação prática materializada no desenvolvimento dum jogo 3D, a desenvolver no decorrer do semestre, usando um motor de jogos. Este trabalho encontra-se faseado em duas entregas ao longo do semestre. Os alunos terão ainda que implementar um conjunto de técnicas para iluminação global partindo dum esqueleto dum ray-tracer.

A cada aluno será atribuída uma nota individual resultante da avaliação dos projetos de programação e de um exame final, sendo neste último necessária a obtenção de nota mínima de 8,00 valores (sem arredondamento). O peso dos projetos será de P1: 25%, P2: 20%, P3: 15%. O exame terá um peso de 40%. Para obter frequência um aluno deverá obter uma média de 9.5 valores na componente laboratorial. Para aprovação, a nota final ponderada deverá ser igual ou superior a 9.5 valores.

4.4.7. Teaching methodologies (including students' assessment):

The topics will be presented with emphasis on its formalisms and motivation. The presentations will be supported by practical demonstrations based on screenshots, demos or interactive applications. The vast majority of the topics will be the subject of practical implementations in a game to be developed throughout the semester using a game engine. This development is evaluated at two distinct times. Students also improve a given ray tracer with a set of global illumination techniques.

Each student is assigned an individual grade based on the 3 programming projects and a final exam. The final exam requires a minimum classification of 8.00 (out of 20).

The weights of each project are: P1 - 25%; P2 - 20%; P3 - 15%. The final exam has a weight of 40%. To be admitted to the exam, the final grade on the programming projects has to be ≥ 9.5 (out of 20). Approval requires a final weighted grade of at least 9.5 (out of 20)

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

Através das aulas teóricas o aluno é iniciado nos conceitos, técnicas, ferramentas matemáticas e algoritmos que permitem a visualização de objetos 3D, com algum grau de realismo e em tempo real, no computador. A avaliação relacionada é baseada em dois testes intermédios.

As aulas práticas são utilizadas principalmente para a conceção e execução dos projetos, podendo os alunos fazer perguntas de esclarecimento. Esses projetos correspondem a várias fases do desenvolvimento de jogos 3D. É incentivada a reutilização de componentes de programação e o promovido o desenho do sistema por forma a que possa ser extensível. A animação por computador, a planificação de movimentos, tratamento de colisões e a simulação física também são objetivos importantes dos projetos.

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

Throughout lectures the student is introduced to the concepts, techniques, mathematical tools and algorithms for real time 3D models visualization featuring some degree of realism. The related evaluation is based on two mid-term tests.

Practical classes are mainly used for the design and implementation of the projects, on which students can ask questions for clarification. These projects correspond to different stages of 3D game development. The reuse of programming components is encouraged as well as an extensible game engine architecture. Computer animation, motion planning, collisions and physics simulation are also important goals for the projects.

4.4.9. Bibliografia de consulta/existência obrigatória:

- Jason Gregory, "Game Engine Architecture", 2nd edition, A K Peters/CRC Press, 2014, ISBN-13: 978-1466560017
 - David H. Eberly, "3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics", 2nd edition, CRC Press, 2006, ISBN-13: 978-0122290633
 - J.F. Hughes, A. van Dam, M. McGuire, D. Sklar, J.D. Foley, S.K. Feiner, K. Akeley, "Computer Graphics: Principles and Practice", 3rd Edition, Addison-Wesley Professional (2013), ISBN-13: 978-0321399526
 - T. Akenine-Moller, E. Haines, N. Hoffman, "Real-Time Rendering", Third Edition, AK Peters (2008), ISBN-13: 978-1568814247
 - R.J. Rost, B.M. Lincea-Kane, D. Ginsburg, J. Kessenich, B. Lichtenbelt, H. Malan, M. Weiblen, "OpenGL Shading Language", 3rd Edition, Addison-Wesley Professional (2009), ISBN-13: 978-0321637635
- Additional readings will be provided during classes and through the course website.*

Mapa IV - Mineração e Pesquisa de Dados Web

4.4.1.1. Designação da unidade curricular:

Mineração e Pesquisa de Dados Web

4.4.1.1. Title of curricular unit:

Web Data Mining and Search

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:

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

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Nuno Manuel Robalo Correia (sem horas de contacto)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***João Miguel da Costa Magalhães - T:28; PL:28***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Conhecimentos*

- *Compreender a representação de informação Web e multimédia*
- *Compreender modelos de representação de Web e multimédia*
- *Analisar dados Web e multimedia para extração de informação*
- *Compreender os paradigmas de acesso a informação Web e multimédia*

Aptidões

- *Implementar modelos de representação de Web e multimédia*
- *Implementar algoritmos de descrição de informação Web e multimédia*
- *Desenvolver sistemas de informação em larga escala para dados Web e multimédia*
- *Compreender as necessidades de informação em domínios específicos*

Competências

- *Desenhar um sistema de recuperação de informação Web e multimodal*
- *Selecionar as técnicas de adequadas à solução de problemas com informação multimodal*
- *Capacidade para desenhar um protocolo experimental e analisar resultados de experimentais*

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

- *Understand the concept of error in information representation.*
- *Understand Web and multimedia representation models.*
- *Analyze Web and multimedia data for information extraction.*
- *Understand Web and multimedia information access paradigms*

Know-how

- *Implement Web and multimedia information representation algorithms.*
- *Implement Web and multimedia information description algorithms.*
- *Develop Web and multimedia information access systems.*
- *Understand domain specific information needs.*

Soft-Skills

- *Design Web and multimodal information retrieval systems.*
- *Select the right techniques do solve problems dealing with multimodal information.*
- *Be able to design experimental protocols and analyze experimental results.*

4.4.5. Conteúdos programáticos:

1. *Introdução*
2. *Representação de dados de media social*
3. *Análise de grafos de redes sociais*
4. *Extração de informação multimodal*
5. *Representações distribuídas de palavras*
6. *Embeddings multimodais*
7. *Anotação de informação de multimodal*
8. *Attention models*
9. *Descrição textual de dados visuais*
10. *Agentes de pesquisa de informação multimodal*
11. *Caso de estudo*

4.4.5. Syllabus:

1. Introduction
2. Social-media data representations
3. Social-networks analysis
4. Multimodal information extraction
5. Distributed representations of words
6. Multimodal embeddings
7. Tagging social-media information
8. Attention models
9. Natural language description of visual data
10. Multimodal conversational search agents
11. Case study

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

A primeira parte do curso discute conceitos base e introduz técnicas e algoritmos elementares de Recuperação de Informação: o modelo de espaço vetorial, relevância, PageRank, indexação, modelos de linguagem e ordenação por aprendizagem.

A parte final do programa, aplica os conceitos base de RI à área de recuperação de informação multimédia e examina os requisitos específicos deste tipo de informação.

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

The first part of the course discusses key concepts and introduces students to well-established techniques and algorithms in Information Retrieval: the vector space model, relevance, PageRank, indexing, language-models and learning to rank.

The final part of the course takes the initial concepts from classic IR and applies them to multimedia information and examines the special requirements of this data.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas é apresentada a matéria, com exemplos e discussão cuidada dos conceitos mais importantes. As aulas laboratoriais destinam-se à realização dos trabalhos de laboratório e o projeto final.

As aulas laboratoriais destinam-se à realização de 4 trabalhos laboratoriais, com duração de duas aulas cada, e do um projeto final que durará 5 semanas.

Será disponibilizada uma página Web da disciplina onde se mantém informação atualizada sobre o funcionamento da mesma. Os slides da matéria teórica e os guias de laboratório estarão disponíveis na página Web da disciplina.

A avaliação da disciplina é composta por 2 testes escritos individuais realizados a meio e no fim do semestre e trabalhos de grupo de laboratório: 4 mini-trabalhos quinzenais e um projeto final.

A nota mínima em cada um dos testes e trabalhos de laboratório é de 8 valores. A componente teórica vale 50%, cada mini-trabalho vale 5% e o trabalho final 30%.

4.4.7. Teaching methodologies (including students' assessment):

The course content is presented in the lectures with practical examples followed by a detailed discussion of the most important concepts. In the laboratory classes support will be given to students to the work assignments and final project.

A web page will be made available with updated information about the course. The teaching materials and laboratory guides will also be available on this web page.

The course grading is composed of two written tests and four laboratory assignments and a final project. The minimum grade of each component is 8.0, the exam weights 50%, each work assignment is 5% and the final project is 30%.

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

A informação presente em dados Web e multimodais permite a criação de sistemas capazes de responder a necessidades de informação bastante variadas e complexas, tais como no domínio médico ou em ambientes de media social. O objetivo deste curso é o de permitir os alunos compreenderem todos os aspetos inerentes à representação, extração e acesso a informação Web e multimodal. Os alunos irão dominar as principais técnicas da criação de espaços de pesquisa multimodais, indexação através de técnicas de hashing e anotação de informação para permitir a pesquisa por texto.

Um projeto permitirá consolidar as variadas técnicas num único sistema de pesquisa Web/multimodal num dado domínio específico, e.g., médico, vídeo vigilância ou media social.

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

The goal of this course is to provide students with an understanding of all aspects of the design and implementation of search engines. Students will master fundamental concepts of Information Retrieval: information representation,

indexing, querying, and ranking by relevance. This first objective is tackled by the first part of the course syllabus where fundamental concepts are introduced. The second part of the syllabus further strengthen these concepts by extending them to multimedia information.

The weekly assignments allow students understand deep concepts and syllabus structure keep students up-to-date with late-breaking research allowing them to pursue an actual research topic.

4.4.9. Bibliografia de consulta/existência obrigatória:

- Bing Liu, "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data", 2013.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", 2016.
- Jazon W Osborne, "Best practices in data cleaning", 2012.

Mapa IV - Modelação de Dados

4.4.1.1. Designação da unidade curricular:

Modelação de Dados

4.4.1.1. Title of curricular unit:

Data Modelling

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Carlos Augusto Isaac Piló Viegas Damásio - T:14; PL:14

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

João Carlos Gomes Moura Pires - T:14; PL:14

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

Saber:

- *Modelos NoSQL*
- *Modelação e interrogação de dados em grafos*
- *Princípios dos Dados Abertos Ligados e conceitos da Web Semântica*
- *Linguagens para representação, raciocínio e interrogação na Web Semântica*
- *Conceitos, arquiteturas e modelos de um Data Warehouse*
- *Modelação multidimensional dos dados para interrogação OLAP*

Saber Fazer:

- *Identificar aplicações que necessitem de modelos não relacionais*
- *Modelar uma base de dados em grafos e interrogá-la (Neo4j com Cypher)*
- *Utilizar um triplestore e motor de inferência (Apache Jena) para interrogar em SPARQL informação existente na Web Semântica*
- *Analisar, construir e interrogar modelos multidimensionais*
- *Utilizar uma base de dados temporal*

Soft-Skills

- *Explorar a literatura atualizada de um tópico*

- Desenvolver o espírito crítico relativamente a tecnologia recente
- Trabalhar em equipa
- Efetuar uma apresentação oral expondo um tema recente

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

Knowledge:

- NoSQL models
- Graph modelling and query languages
- Linked Open Data principles and Semantic Web concepts
- Languages for representing, reasoning and querying in the Semantic Web
- Concepts, architectures and models of a Data Warehouse
- Multidimensional data modelling for OLAP querying.

Application:

- Identify applications requiring non relational modelling
- Model a graph database and query it (Neo4j with Cypher)
- Use a triple store and inference engine (Apache Jena) for querying with SPARQL data in the Semantic Web
- Analyse, design and query multidimensional models.
- Use a temporal database

Soft-Skills

- To explore autonomously the recent bibliography
- To develop critical reasoning regarding recent technology
- To work in a team
- To orally present a survey on a recent topic

4.4.5. Conteúdos programáticos:

1. Modelos de dados NoSQL

Modelos alternativos para grandes volumes de dados. Movimento NoSQL. Dados relacionais, semiestruturados e em grafos. Modelação de informação em colunas, documentos e grafos. Linguagens de consulta para modelos em grafos. Bases de dados em grafos. Relação e comparação com outros modelos do movimento NoSQL e armazenamento de dados chave-valor.

2. Web Semântica

Motivação. Dados abertos ligados. Linguagem e semântica do Resource Description Framework (RDF) e linguagem de consulta SPARQL. Representação de ontologias na Web Semântica: RDF Schema e Web Ontology Language (OWL).

3. Processamento Analítico em Linha (OLAP)

Armazéns de dados. Modelos (conceptuais) multidimensionais. Operações base e linguagens de interrogação OLAP. Metadados. Dimensões espaciais e temporais. Interatividade na análise de dados.

4. Exercícios de Desenvolvimento e Projecto Final

Utilização de ferramentas (graph database, bases de dados temporais, RDF e OWL API OLAP e multidimensionais)

4.4.5. Syllabus:

1. NoSQL data models

Alternative models for storing big volumes of data. Column, document and graph models. Relational, semi-structured and graph data. Data modelling with graphs. Querying graph models. Graph databases. Relationship to NoSQL movement and key-value stores.

2. Semantic Web

Motivation. Linked Open Data. Language and semantics of the Resource Description Framework (RDF) and SPARQL query language. Ontologies in the Semantic Web: RDF Schema and Web Ontology Language (OWL).

3. Online Analytical Processing (OLAP)

Data Warehouses. (Conceptual) multidimensional data models. Typical OLAP operations and OLAP query languages. Metadata. Spatial and temporal dimensions. Interaction in the data analysis process.

4. Exercises and final project

Use of tools (graph database, temporal databases, RDF and OWL API, OLAP and multidimensional)

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

Esta UC aborda os modelos existentes e linguagens de consulta para dados semanticamente enriquecidos na Web, aplicações com grandes volumes de dados e de dados abertos. Concentra-se nos modelos baseados em grafos e modelos multidimensionais com tempo espaço.

Existe uma correspondência imediata entre os conteúdos e os objetivos. Todos os conhecimentos teóricos são complementados com experiência prática em modelação e utilização de linguagens e ferramentas.

A UC apresenta uma estrutura lógica das matérias de acordo com os manuais de referência, abordando os “saberes” nas aulas teóricas (pontos 1 a 3 dos conteúdos programáticos).

Os “saber-fazer” são obtidos nas aulas práticas onde são exploradas autonomamente aplicações e ferramentas na resolução de tarefas e problemas dados (essencialmente, ponto 4 do conteúdo programático).

As competências são adquiridos com a realização do trabalho prático (essencialmente, ponto 4 do conteúdo programático).

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

This course address data modelling and query languages for semantically enriched data, big data and/or open data applications in the Web. The core of the course covers in detail graph and multidimensional modeling, with time and space.

There is an immediate correspondence between the syllabus and the curricular unit’s objectives. The topics covered in

the recitation lectures are complemented with hands-on experience of modelling, use of languages and tools. This course presents an up-to-date view of the subjects, following a logical structure according to the reference text books, addressing the knowledge learning outcomes in the recitation lectures (items 1 to 3 of syllabus) The “knowledge application” objectives are obtained in the lab classes where students explore tools and applications for solving given specific tasks and problems (mostly in item 4 of syllabus). The competences are acquired with the practical project work (mostly in item 4 of syllabus).

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas são expostos os tópicos centrais que introduzem o aluno aos conceitos principais da unidade curricular que lhe permitirão depois aprofundarem algum tema em particular, acompanhados de acetatos teóricos especialmente desenvolvidos para a UC.

Os alunos deverão entregar um trabalho prático sobre um tema, à sua escolha. Os alunos deverão autonomamente explorar a informação existente na literatura, sendo a bibliografia inicial discutida com o docente.

As aulas práticas destinam-se à exploração de diversas ferramentas que permitam a sua experimentação e utilização das linguagens lecionadas, resolvendo tarefas predefinidas.

A avaliação da unidade curricular é constituída por 2 testes individuais (cada 25%), um projeto/trabalho final de grupo (35% da nota), apresentação oral e discussão de um trabalho de um colega (15% da nota).

A nota mínima em cada teste é de 8 de valores e média dos testes deverá ser superior ou igual a 10 valores, após arredondamento.

4.4.7. Teaching methodologies (including students' assessment):

In the recitation lectures (2T) the main subjects are exposed and laboratory classes (2P). These are intended to be autonomously explored by students. There are written lecture notes and hand out slides, which closely follow the presentation in recitation lectures.

In the laboratory classes students explore tools that allow experimenting their use, as well as use the query languages for solving predefined problems and tasks.

Evaluation consists in 2 individual midterms (each worth 25% of the final grade), a final team project (35%), oral presentation and discussion of a colleague's project (15% of the grade).

Each midterm has a minimum grade of 8/20 and the average of the midterms must be at least 10/20, after rounding.

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

A unidade aborda áreas de conhecimento muito recente e em constante evolução, consequentemente as metodologias de ensino estão concebidas para dotar o aluno com os fundamentos teóricos e práticos de modelação de dados avançada, estimulando a sua autonomia e espírito crítico.

Os tópicos elencados nos “saberes” são tratados nas aulas teóricas da unidade curricular, sequenciada de acordo com a estrutura lógica dos manuais adotados e sendo disponibilizados acetatos. Todos os módulos da unidade curricular apontam para leituras complementares de artigos científicos disponibilizados na página Web da cadeira ou disponíveis na Web, assim como apontadores para as recomendações do W3C. É dada especial ênfase aos algoritmos de inferência, capacidades expressivas de cada uma das linguagens, limitações e complexidade temporal para estimular a discussão e identificar claramente os compromissos.

Os objetivos da componente “Saber-fazer” são garantidos com as aulas práticas, onde os alunos contactam com ferramentas e metodologias disponíveis, cujos algoritmos de base foram estudados nas aulas teóricas. As ferramentas são instaladas pelos alunos em diversos sistemas de operação. As ferramentas são utilizadas na resolução de tarefas e problemas desenhados pelo docente de forma a ilustrar os aspetos mais salientes e distintivos, assim como dificuldades que podem encontrar na construção de aplicações reais. As aulas práticas são também utilizadas para realizar exercícios preparatórios para os testes. Os testes realizados durante o semestre garantem que os alunos seguem a matéria e se encontram preparados para a realização do projeto/trabalho final.

O projeto/trabalho final é uma componente essencial da avaliação possibilitando a aquisição das “Competências” identificadas nos objetivos da unidade curricular. O trabalho é realizado em grupos de 2 ou 3 elementos, versando um tópico proposto e identificado pelos alunos a partir da literatura científica da área e que normalmente aprofunda alguma parte da matéria teórica complementando com uma aplicação prática. Neste processo são acompanhados pelo docente, discutindo os tópicos possíveis e indicando bibliografia de base para o trabalho. Esta discussão do tema de trabalho é realizada parcialmente nas aulas práticas e durante o horário de atendimento do docente. O trabalho é redigido preferencialmente em língua inglesa e apresentado em inglês num “workshop” da unidade curricular no final do semestre.

Adicionalmente, os alunos avaliam anonimamente um projeto dos outros colegas de acordo com formulários de revisão utilizados em conferências científicas. Os formulários de avaliação dos discentes são também classificados e integrados no cômputo da nota final.

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

The unit addresses recent areas of computer science and in fast evolution, consequently the teaching methodologies are conceived to provide the accepted theoretical and practical fundamentals of Advanced Data Modelling but at the same time stimulate autonomy and critical spirit.

The “Knowledge” topics are addressed in the lectures, according to the logical sequence of the reference manuals and slides are made available. All the modules of the course provide pointers to scientific papers with complementary readings, and the W3C recommendations. We focus in the inference algorithms, expressive capabilities of the languages, limitations and computational complexity in order to stimulate discussion and clearly identifying the compromises.

The “application” outcomes are attained in the lab classes, where the students use the tools and methodologies, using

the algorithms exposed in the recitation lectures. The tools are installed by the students in different operating systems, and are used to solved tasks and problems designed by the lecturer to illustrate the most important and distinctive issues, as well as the difficulties that can be found in the development of real applications. The lab classes are also used to solve preparatory exercises for the midterms, e.g. for understanding the major languages or algorithms.

The midterms guarantee that the students do study at the appropriate time the subjects in order to be able to develop their final project. The final project is an essential component of the evaluation allowing for the achievement of the competences learning outcomes. The project is developed by teams of 2 or 3 students about a topic proposed and identified by them. The topic is proposed and identified by the students by analysing the scientific literature of the area, and that usually expands part of the subjects studied in the classes complemented with some application or practical work. In this process, the students are accompanied by the lecturer by discussing the topics and helping them in the identification of the basic literature. The discussion of the topic is partially performed in the lab classes and in the office hours. The work is preferably written in English and presented in a course's "workshop" at the end of the semester.

Additionally, the students review other colleagues' projects using typical conference review forms. The reviews are classified by the lecturer and used to obtain part of the final grade.

4.4.9. Bibliografia de consulta/existência obrigatória:

- *Ian Robinson, Jim Webber, and Emil Eifrem. Graph Databases. O'Reilly Media, Inc, 2013.*
- *Grigoris Antoniou, Paul Groth, Frank van Harmelen and Rinke Hoekstra . A Semantic Web Primer, 3rd Edition. MIT Press, August 2012.*
- *The Description Logic Handbook. Theory, Implementation and Applications. Edited by Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi and Peter Patel-Schneider. Cambridge University Press, June 2010.*
- *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Third Edition) - Ralph Kimball, Margy Ross. Wiley, 2013.*
- *Guy Harrison. Next Generation Databases: NoSQL, NewSQL and Big Data. Apress, 2015. ISBN: 978-1484213308.*
- *Dan Sullivan. NoSQL for Mere Mortals. Addison-Wesley, 2015. ISBN: 978-0134023212*
- *Ted Hills. NoSQL and SQL data modeling. Technics Publications, 2016. ISBN: 978-1634621090*

Mapa IV - Processamento de Streams

4.4.1.1. Designação da unidade curricular:

Processamento de Streams

4.4.1.1. Title of curricular unit:

Stream Processing

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Nuno Pregoça (sem horas de contacto)

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

João Carlos Gomes Moura Pires - T:14; PL:14
Sérgio Marco Duarte - T:14; PL:14

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

Aprender os conceitos fundamentais, linguagens e sistemas para construir aplicações que processam fluxos de dados. A unidade curricular aborda, apresenta e discute os sistemas generalistas para processamento de streams em tempo real, e desenvolverá com especial ênfase o estudos de sistemas para modelos de dados estruturados orientados aos fluxos de dados.

Saber

- A. Conhecer os principais modelos de programação para o processamento de fluxos contínuos de dados
- B. Conhecer as linguagens e assimilar as características fundamentais para resolver problemas no domínio do processamento de streams
- C. Compreender as vantagens e as desvantagens das plataformas para processamento de streams.

Aplicação

- A. Ser capaz de escolher os modelos, linguagens e ferramentas mais adequadas para resolver um problema de processamento de streams
- B. Ser capaz de desenvolver e executar aplicações para processamento de streams usando ferramentas e tecnologias correntes.

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

Learn the fundamental concepts, languages, and systems for building applications that process data streams. This course discusses, presents and discusses generalist systems for real-time stream processing, and will focus on the study of systems for structured data flow-oriented models.

To knowledge

- A. Know the main programming models for streaming data processing
- B. Know the languages and assimilate the fundamental characteristics to solve problems in the stream processing domain.
- C. Understand the advantages and disadvantages of stream processing platforms.

Application

- A. Being able to choose the most appropriate models, languages and tools to solve a stream processing problem.
- B. Set capable of developing and executing stream processing applications using current tools and technologies.

4.4.5. Conteúdos programáticos:

1.Data Stream Management Systems (DSMS)

- Modelos estruturas de dados para streams.
- Operações algébricas sobre streams e relações (continuous queries, agregados, janelas temporais).
- Linguagens de continuous queries.
- Extensões ao SQL e sistemas de bases de dados para lidar com streams.

2.Processamento de Eventos

- Streams como sequências de eventos.
- Regras de produção, regras reativas e computação guiada por eventos.
- Redes de processamento de eventos, agentes e canais.
- Eventos complexos e derivados. Detecção de padrões de eventos.
- Linguagens e sistemas para processamento de eventos.

3.Sistemas distribuídos para processamento de streams

- Modelos para processamento de streams: streams como sequências de mini-batches (e.g. Spark streaming); processamento contínuo (e.g. Apache Flink, Storm)
- Modelos de programação.
- Aspectos de sistemas: distribuição, escalabilidade e tolerância a falhas.
- Bases de dados distribuídas de séries temporais.
- Sistemas para processamento de streams IoT

4.4.5. Syllabus:

1. Data Stream Management Systems (DSMS).

- Structured Data Models for Streams.
- Algebraic operators on stream and relations (continuous queries, aggregates and blocking, time windows).
- Continuous query languages.
- Languages and systems that extend SQL and database management systems to deal with data streams.

2. Complex Event Processing.

- Streams as sequences of events.
- Production rules, reactive rules, and event-driven computing.
- Event processing networks, agents and channels.
- Complex and derived events. Detection of event patterns.
- Event-processing languages and systems.

3. Distributed Stream Processing Systems.

- System models for stream processing: streams as sequences of mini-batches (e.g. Spark streaming); continuous processing (e.g. Apache Flink, Storm).
- Programming models.
- System aspects: distribution, scalability and fault-tolerance.
- Distributed time-series database.
- Systems for IoT stream processing.

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

Cada um dos grandes tópicos do programa fornece os fundamentos de uma abordagem a processamento de streams, e inclui exemplos de linguagens e sistemas que permitem aos estudantes consolidar os conceitos e construir pequenas aplicações. A combinação dos grandes tópicos permite uma visão sistémica que vai das componentes mais baixo nível, de conceção de sistemas computacionais, até à de mais alto nível, construídas sobre estes, com linguagens declarativas para processamento de dados de stream.

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

Each of the syllabus main topics provides the fundamentals of an approach to process data streams, and includes examples of languages and system that allow the student to consolidate the concepts and build small applications. The combination of the main topics provides a systemic view ranging from the more low level components of computer systems, to high level components built on top of those, with declarative languages to process data streams.

4.4.7. Metodologias de ensino (avaliação incluída):

As aulas teóricas apresentam os modelos, linguagens e plataformas dominantes para cada área temática do programa, usando exemplos paradigmáticos para enquadrar a discussão e o entendimento das questões tratadas.

As aulas práticas focam-se na utilização prática dos sistemas e linguagens estudadas, incluindo o desenvolvimento de aplicações para cenários particulares de processamento de streams. As aulas consistem em demonstrações, exercícios e o acompanhamento da realização dos dois projectos de programação a desenvolver ao longo do semestre. Exemplos de tais aplicações, que têm sido usados, são os challenges da principal conferência nesta área (DEBS).

A avaliação consiste em dois testes (20% + 40%) e dois trabalhos práticos (20% cada).

4.4.7. Teaching methodologies (including students' assessment):

Lectures present the dominant models, languages and platforms for each thematic area of course syllabus, using paradigmatic examples to frame the discussion and the understanding of the issues.

Labs promote hands-on experience on the covered systems and languages, including application development for particular data streaming scenarios. Classes comprise demos, exercises and support for two programming projects to be developed during the semester. Examples of such application are the challenges of the main conference in this area (DEBS).

Grading comprises two quizzes (20% and 40%) and two project assignments (20% each).

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

Os tópicos necessários para atingir os objetivos estabelecidos para a unidade curricular são ministrados nas aulas teóricas e práticas. A aquisição destes conhecimentos é avaliada nas provas escritas (testes/exames). A componente de projeto garante que no final os estudantes conseguem de facto construir aplicações nesta área.

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

The topics required to achieve the objectives established for the course are taught in the lectures and labs. The acquisition of knowledge is assessed in written midterms. The project component assures that students are indeed able to build applications in this area.

4.4.9. Bibliografia de consulta/existência obrigatória:

*Opher Etzion and Peter Niblett. Event Processing in Action. Manning Publications, 2011.
Lukasz Golab and Tamer Özsu. Data Stream Management. Morgan and Claypool, 2010.
Selection of papers.*

Mapa IV - Programação com Restrições

4.4.1.1. Designação da unidade curricular:

Programação com Restrições

4.4.1.1. Title of curricular unit:

Constraint 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:***168***4.4.1.5. Horas de contacto:***T:28; PL:28***4.4.1.6. ECTS:***6***4.4.1.7. Observações:***Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Pedro Manuel Corrêa Calvente Barahona - T:14; PL:14***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Jorge Carlos Ferreira Rodrigues da Cruz - T:14; PL:14***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Saber:*

- *Problemas de satisfação de restrições, e sua complexidade combinatória.*
- *Domínios discretos (finitos e booleanos) e contínuos. O caso SAT.*
- *Propagação de restrições e tipos de consistência.*
- *Algoritmos para manutenção de consistência.*
- *Heurísticas.*

Saber Fazer:

- *Modelação e resolução de problemas.*
- *Seleção das restrições e algoritmos mais adequados.*
- *Escolha de heurísticas eficientes.*

Competências Complementares:

- *Compreensão de especificações informais.*
- *Identificação dos modelos mais apropriados.*
- *Capacidade de abstração e generalização.*
- *Análise e explicação de resultados.*
- *Capacidade de pesquisa de literatura*

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

- *Constraint Satisfaction Problems, and its combinatorial complexity.*
- *Discrete (finite and Boolean) and continuous domains. The special case of SAT.*
- *Constraint propagation and types of consistency.*
- *Algorithms for consistency maintenance.*
- *Heuristics.*

Application:

- *Problem modelling and solving.*
- *Selection of the most adequate constraints and algorithms.*
- *Choice of efficient heuristics.*

Soft-skills:

- *Understanding of informal specifications.*
- *Identification of the most adequate models.*
- *Abstraction and generalisation skills.*
- *Analysis and explanation of results.*
- *Specialised literature search*

4.4.5. Conteúdos programáticos:

1. *Problemas de decisão em domínios discretos.*
2. *Modelação de Problemas.*
3. *Resolução de Problemas.*
4. *Introdução aos problemas de restrições em domínios contínuos.*
5. *Restrições e Análise de Intervalos.*
6. *Método de Newton para Intervalos*
7. *Associação de Funções de Redução de Domínio a Restrições.*
8. *Propagação de Restrições e Manutenção de Consistência.*
9. *Resolução de problemas em domínios contínuos.*

4.4.5. Syllabus:

1. *Decision problems in discrete domains.*
2. *Problem Modelling.*
3. *Problem Solving.*
4. *Introduction to Interval Constraints.*
5. *Continuous Constraints and Interval analysis.*
6. *Interval Newton Method.*
7. *Associating Narrowing Functions to Constraints.*
8. *Constraint Propagation and Consistency Enforcement.*
9. *Problem Solving in Continuous Domains.*

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

Nesta unidade curricular aprofundam-se os conhecimentos de pesquisa referidos como opcionais no ACM 2008 CS Curriculum na área de Sistemas Inteligentes (IS/AdvancedSearch [elective]), e que complementam os introduzidos em unidades anteriores.

A resolução de problemas complexos de decisão (modelados com restrições) tem sido muito melhorada pelos avanços recentes na área de programação por restrições, não apenas em domínios discretos (incluindo booleanos), mas também em domínios contínuos, nomeadamente através da redução do espaço de pesquisa através da propagação de restrições.

A matéria leccionada insere-se claramente nos objetivos de dar a conhecer aos estudantes os avanços recentes mas já consolidados nesta área, que lhes permitam identificar oportunidades de aplicação destas técnicas em variadas situações práticas, modelar os problemas identificados e resolvê-los de forma eficiente.

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

This course addresses in more depth the topics of search referred to as elective in the ACM 2008 CS Curriculum in the area of Intelligent Systems (IS/Advanced Search [elective]), which complement those introduced in previous courses. Solving complex decision problems (modelled with constraints) have undergone significant improvements in recent years, taking advantage of corresponding advances in constraint programming, both in finite (including Boolean) and continuous domains, namely through the search space reduction by means of constraint propagation.

The syllabus clearly fulfil the objectives of the course, which is meant to provide the students with the advanced, if consolidated, knowledge in this area, allowing them to detect opportunities to apply it in many situations, modelling the identified problems, and efficiently solve them.

4.4.7. Metodologias de ensino (avaliação incluída):

O programa é leccionado em aulas teóricas e práticas. Nas primeiras são leccionados os conceitos e técnicas relevantes bem como a forma como estão implementados nas linguagens e nos sistemas de modelação e resolução utilizados. Nas aulas práticas são resolvidos problemas retirados de benchmarks típicos destas áreas, sendo explorados e desenvolvidos modelos e técnicas de pesquisa apropriados.

A avaliação de conhecimentos, inclui:

- *2 testes individuais teóricos (domínios finitos e domínios contínuos).*
 - *2 trabalhos práticos sobre a resolução de problemas de média dimensão.*
- A nota final é a média das 4 componentes, sendo necessária uma nota não inferior a 8.0 valores quer na média dos 2 testes, quer na média dos dois trabalhos prático. A nota dos testes teóricos pode ser substituída pela nota no exame final.*

4.4.7. Teaching methodologies (including students' assessment):

The syllabus is taught in theoretical and laboratory classes. In the former, the main concepts and techniques are addressed, together with their implementation in the modelling and execution languages and systems adopted. The laboratory classes are dedicated to solving problems, selected from typical benchmarks in these áreas, in order to exploit and develop adequate models and search techniques.

The assessment and grading of students includes:

- *2 mini tests (on discrete and finite domains)*
- *2 practical projects on resolution of mid-sized constraint problems.*

The final grade is the average of these 4 components, a minimum grade of 8.0 being required both in the average of the 2 mini-tests and in the average of the practical projects. The grade of the theoretical tests can be replaced by the grade obtained 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 a) a aquisição de conhecimento recentemente desenvolvido para a satisfação de problemas de restrições; b) a experimentação da resolução de problemas, com base em modelos e

algoritmos de pesquisa alternativos; e c) avaliação da adequação e eficiência das soluções alternativas, identificando-se as mais apropriadas para diferentes situações.

O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição dos vários conceitos e técnicas de programação de restrições, sendo estudadas as suas principais propriedades (por exemplo, complexidade). Estas matérias são avaliadas em testes individuais, onde os alunos têm um tempo limitado para mostrar o seu grau de assimilação das matérias lecionadas.

O comportamento de diferentes modelos e técnicas de pesquisa a aplicações concretas, requer uma aprendizagem mais baseada na experimentação, já que resultados teóricos existentes (por exemplo, o estudo da complexidade no pior caso) são pouco informativos. Para esse efeito, essa experimentação é feita nas aulas práticas em problemas razoavelmente simples, sendo avaliada através de trabalhos em aplicações não triviais, e que obrigam os estudantes a desenvolver capacidades de análise e de decisão bastante importantes nesta área.

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

The objectives of this course identify a) the acquisition of recently developed knowledge to solve constraint satisfaction problems; b) the experience obtained in solving such problems, adopting alternative models and search techniques; and c) assessment of the adequacy and efficiency of alternative approaches, with the identification of the most appropriate for different situations.

Objective a) is basically achieved in the theoretical classes, where the main concepts and techniques of constraint programming are explained, together with the study of their main characteristics (e.g. complexity). These topics are assessed in 2 individual mid/final-term tests, where the students have a limited time to show their level of assimilation of the relevant knowledge.

Such time limitation is inadequate to study the application of different models and techniques, which requires a more experiment-oriented training, since most theoretical results available (e.g. worst case complexity) have limited application. For this purpose, practical experimentation with simple problems is done in the laboratory classes, and is assessed in more complex projects, requiring the students to develop analysis and decision skills of great relevance in this area.

4.4.9. Bibliografia de consulta/existência obrigatória:

Main References:

1. Dina Rechter, *Constraint Processing*, Morgan Kaufman, 2003.
2. Krzysztof Apt, *Principles of Constraint Programming*, Cambridge University Press, 2009 (online)
3. Jaulin, L., Kieffer, M., Didrit, O., Walter, E., *Applied Interval Analysis*, Springer, 2001.
4. Eldon Hansen, G. William Walster, *Global Optimization Using Interval Analysis*, Marcel Dekker, 2003
5. Jorge Cruz, *Constraint Reasoning for Differential Models*, 126 *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2005.

Complementary Reference:

1. Comet Tutorial, *Dynamic Decision Technologies Inc.*, August 28, 2009.
2. *Handbook on Constraint Programming*, (F. Rossi, P. van Beek and T. Walsh, eds.), Elsevier, 2006
3. *Handbook of Satisfiability*, (A. Biere, M. Heule, H. Van Maaren and T. Walsh, eds.), IOS Press, 2009
4. Ramon E. Moore, *Interval Analysis*, Prentice-Hall, 1966

Mapa IV - Programação Concorrente: Linguagens e Técnicas

4.4.1.1. Designação da unidade curricular:

Programação Concorrente: Linguagens e Técnicas

4.4.1.1. Title of curricular unit:

Concurrent Programming: Languages and Techniques

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:*Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***António Maria Lobo César Alarcão Ravara - T:14; PL:7***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Carla Maria Gonçalves Ferreira - T:14; PL:7**Bernardo Parente Coutinho Fernandes Toninho - PL:14***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***O objetivo central da UC é ensinar a desenvolver programas concorrentes para construir aplicações realistas, usando linguagens mainstream avançadas como Erlang, Go e Rust.**No final desta unidade curricular o estudante terá adquirido conhecimentos, aptidões e competências que lhe permitam:*

- *Compreender os desafios da programação concorrente e os conceitos e técnicas essenciais para desenvolver aplicações corretas;*
- *Ser capaz de desenvolver soluções típicas para lidar com os desafios usuais, argumentando e/ou mostrando a sua correção;*
- *Conhecer as linguagens modernas para desenvolvimento de programas concorrentes e os padrões principais para lidar com os aspetos críticos de uma aplicação concorrente.*

4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):*The main goal of this unit is to teach the students the development techniques that are needed for the development of concurrent programs in a realistic, general purpose setting, using advanced mainstream languages such as Erlang, Go and Rust.**By the end of this unit, students will have acquired knowledge, skills and competences that will enable them to:*

- *Understand the challenges of concurrent programming and the concepts and fundamental techniques required to the development of correct applications;*
- *Be able to develop standard solutions to address common challenges in this area, arguing for or showing their correctness;*
- *Know how to instantiate the main design patterns for concurrent programming using state-of-the-art programming languages and handle the critical aspects of a concurrent application.*

4.4.5. Conteúdos programáticos:**1.Programação Concorrente por troca de mensagens***-Conceitos e modelos**-A linguagem Go**-Programação concorrente em Go**a)Canais síncronos**b)hreads verdes (goroutines) e Select**c)Assincronia**d)Padrões de programação concorrentes**-Ferramentas de apoio ao desenvolvimento de aplicações por troca de mensagens em Go**a)Detectores de corridas e bloqueios em tempo de execução**b)Model-Checking***2.Programação Concorrente por controle de permissões***-Conceitos e modelos**-A linguagem Rust**-Programação concorrente em Rust**a)Canais e troca de mensagens**b)Memória partilhada, Arcs e Mutexes**c)Programação com eventos assíncronos**d)Ferramentas de apoio ao desenvolvimento em Rust***3.Programação concorrente com actores***-Conceitos e modelos**-A linguagem Erlang**-Programação concorrente em Erlang**a)Processos**b)Troca de mensagens**c)Padrões de desenho de processos (OTP)**d)Robustez (tratamento de erros, time-outs)**-Ferramentas de apoio ao desenvolvimento em Erlang**a)Desenvolvimento de testes**b)Model checking*

4.4.5. Syllabus:**1. Concurrent Programming using message passing**

-Models and concepts

-The Go programming language

-Concurrent programming in Go

a) Synchronous Channels

b) Green threads (goroutines) and select

c) Asynchrony

d) Concurrent programming patterns

-Tools and techniques to support message passing concurrent programming in Go

a) Runtime race and deadlock detection

b) Model checking

2. Concurrent programming using ownership and permissions

-Models and concepts

-The Rust programming language

-Concurrent programming in Rust

a) Channels and message passing

b) Shared memory, arcs and mutexes

c) Asynchronous event-driven programming

-Tools and techniques to support message passing concurrent programming in Rust

3. Concurrent programming using Actors

-Models and concepts

-The Erlang language

-Concurrent programming in Erlang

a) Processes

b) Message passing

c) Process design patterns (OTP)

d) Robustness (error handling, time-outs)

-Tools and techniques to support concurrent programming in Erlang

a) Model checking

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

As linguagens selecionadas são usadas para desenvolver aplicações e sistemas hoje em dia muito relevantes. Não só são adaptadas para lidar com os principais desafios que se colocam ao fazer programação concorrente (controle de interferência, ausência de bloqueios, etc), como refletem e incorporam desenvolvimentos recentes da investigação em programação concorrente. Ficam assim os alunos a conhecer linguagens mainstream modernas, populares e adequadas ao desenvolvimento de aplicações concorrentes, que oferecem construções capazes de ajudar a obter o comportamento desejado. A natureza avançada das linguagens selecionadas e as suas ligações a conceitos e temáticas de investigação em programação concorrente permitem também estabelecer ligações entre a investigação de ponta em programação concorrente e a sua realização em conceitos e construções numa linguagem de programação concreta e de uso industrial.

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

The selected languages are used in industrial practice to develop applications and systems that are extremely relevant. Not only are these languages suited to address the main challenges that concurrent programmers face (controlling interference, deadlock freedom, etc.), but they also reflect and incorporate recent developments from concurrent programming research. Students are thus introduced to mainstream, industrial-strength languages that are suited for the development of concurrent applications by providing constructs and abstractions that are able to produce the desired behaviors. The advanced and cutting edge nature of the selected languages, combined with their connections to concepts and themes from concurrent programming research also enable the connection between research in concurrent programming and its materialization in concepts, techniques and abstractions in modern, real-world programming languages.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas são apresentados os problemas típicos, os conceitos centrais para caracterizar e lidar com os problemas; as linguagens modernas que fornecem construções e abstrações adequadas a lidar com os problemas; soluções típicas para os problemas; as ferramentas e técnicas de apoio ao desenvolvimento de aplicações correctas e a sua ligação a tópicos de investigação em programação concorrente.

Nas aulas práticas os alunos desenvolvem pequenos programas para ilustrar o uso das técnicas aprendidas e para explorar o uso de linguagens e técnicas de programação concorrente modernas. Estes programas permitem incrementalmente desenvolver uma aplicação realista.

Em trabalho autónomo os alunos realizam um projecto em cada uma das linguagens, usando as ferramentas apresentadas, para ganhar experiência de desenvolvimento.

A avaliação tem duas componentes, cada uma para 50% da nota final: os trabalhos das aulas e os projectos.

4.4.7. Teaching methodologies (including students' assessment):

The lectures will introduce the key challenges and problems that arise in concurrent programming; the main concepts that characterize and address these challenges; the modern programming languages that provide abstractions and constructs that adequately deal with these issues; the standard ways to address concurrency problems; tools and techniques that assist the development of correct concurrent applications and their connection to concurrent programming research.

The laboratory classes will enable students to develop small programs that illustrate the use of the learned techniques and allow them to explore modern concurrent programming techniques. These programs will enable the incremental development of a realistic application.

In autonomous work students will develop a project in each of the languages, using the techniques introduced in this course.

The assessment has two components, each for 50% of the final grade: the exercises done in the classes and the projects.

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

A apresentação dos conceitos nas aulas teóricas permite que os alunos compreendam os desafios fundamentais da programação concorrente e as técnicas gerais de lidar com estes desafios. A introdução de linguagens de programação concorrente modernas permite que os alunos possam aprender a materializar estas técnicas utilizando as abstrações e construções fornecidas por estas linguagens avançadas, abordando assim os problemas da programação concorrente de forma prática e concreta e com o objetivo fundamental do desenvolvimento de programas robustos, correctos e fiáveis em contextos novos e desafiantes.

A resolução de exercícios de modelação e de programação nas aulas práticas permite aos alunos tornarem-se capazes de desenvolver aplicações corretas nas linguagens apresentadas nas aulas teóricas, instanciando os conceitos e as técnicas em problemas e soluções concretas e tirando partido das construções e paradigmas de programação concorrente presentes em cada linguagem. A ênfase em três linguagens com modelos de programação concorrente diferentes permite que os alunos possam explorar de forma hands on a adequação dos vários modelos de programação concorrente à resolução eficaz de classes diferentes de problemas.

Os projetos permitem que os alunos amadureçam as suas competências como programadores de aplicações concorrentes, compreendendo os prós e contras bem como áreas alvo presenciais de cada linguagem.

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

The presentation of the concepts in lectures enables students to understand the fundamental challenges of concurrent programming and the general techniques that can address these challenges. The introduction of modern concurrent programming languages allows students to learn how to instantiate these techniques by employing the abstractions and constructs that are found in these advanced languages, approaching the problems of concurrent programming in a practical and concrete way, guided by the fundamental goal of developing robust, correct and reliable programs in challenging new settings.

The modelling and programming exercises of the laboratory classes allow students to be able to develop correct applications in the languages introduced during lecture, instantiating the concepts and techniques in concrete problems and solutions, leveraging the paradigms and concurrent programming constructs of each language. The emphasis in three distinct languages with different concurrent programming models allows students to explore in a hands on way the adequacy of the various features in effectively solving different classes of problems.

The projects allow students to mature their skills as concurrent programs, understanding the trade-offs and the target domains of each language and programming model.

4.4.9. Bibliografia de consulta/existência obrigatória:

Programming languages:

- Alan A. A. Donovan, Brian W. Kernighan. *The Go Programming Language*
- Steve Klabnik, Carol Nichols. *The Rust Programming Language*
- Joe Armstrong. *Programming Erlang: Software for a Concurrent World*

Foundations of concurrent programming:

- Robin Milner. *Communicating and Mobile Systems: The Pi-Calculus*
- Dave Clarke, Johan Ostlund, Ilya Sergey, and Tobias Wrigstad. *Ownership Types: A Survey*
- Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, and Derek Dreyer. *RustBelt: Securing the Foundations of the Rust Programming Language*
- Gul Agha. *Actors: a model of concurrent computation in distributed systems*

Complementary references:

- William Kennedy with Brian Ketelsen and Erik St. Martin. *Go in Action*
- Fred Hebert. *Learn you some Erlang for great good!*

Mapa IV - Prospeção e Análise de Dados

4.4.1.1. Designação da unidade curricular:

Prospeção e Análise de Dados

4.4.1.1. Title of curricular unit:*Data Analytics and Mining***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:28; PL:28***4.4.1.6. ECTS:**

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Pedro Manuel Corrêa Calvente Barahona (sem horas de contacto)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Susana Maria Santos Nascimento Martins Almeida - T:14; PL:14**Joaquim Francisco Ferreira da Silva - T:14; PL:14***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Conhecimentos:*

- *Pré-processamento de dados*
- *Métodos de redução de dimensionalidade*
- *Algoritmos de agrupamento (crisp e fuzzy) para dados entidade-atributo e grafo*
- *Métodos de validação e interpretação de agrupamento*
- *Métodos de Extração de Informação Relevante*
- *Análise simbólica e estatística de textos*
- *Descritores e classificação de Documentos*
- *Casos de estudo em Data Analytics e Text Mining*

Aptidões:

- *Implementar / adaptar algoritmos de Data Analytics e Text Mining.*
- *Modelar experimentalmente dados reais*
- *Avaliar e interpretar resultados experimentais.*

Competências:

- *Capacidade de escolher e avaliar a adequação dos métodos a casos de estudo*
- *Capacidade de abstração e de generalização*
- *Capacidade de análise crítica*
- *Autonomia em aplicar e aprofundar conhecimentos adquiridos*

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

- *Data pre-processing*
- *Dimensionality reduction*
- *Clustering algorithms (crisp and fuzzy) for entity-attribute and graph-data*
- *Cluster validation and interpretation methods*
- *Relevant Information extraction methods*
- *Symbolic analysis and statistical analysis of text*
- *Document descriptors and document Classification*
- *Case studies in Data Analytics and Text Mining*

Skills

- *Implement / adapt Data Analytics and Text Mining algorithms;*
- *Model real data experimentally.*
- *Assessment and interpretation of experimental results.*

Competences

- *Hability to choose and evaluate the suitability of methods to case studies*
- *Abstraction and generalisation skills*
- *Critical analysis skills*
- *Autonomy and self-reliance in the application and furthering studies*

4.4.5. Conteúdos programáticos:

Introdução

Data Analytics: perspetivas e exemplos de aplicação

Prospecção de Informação estruturada ou não estruturada

Data Analytics

- *Compreender os dados*
- *Pré-processamento de dados*
- *Redução de Dimensionalidade*

Análise de Componentes Principais

Decomposição em valores singulares

- *Modelação Descritiva (crisp /fuzzy)*

Algoritmos de agrupamento por pontos representativos

Algoritmos de agrupamento espectral

- *Validação e Interpretação de Modelos Descritivos*
- *Casos de Estudo*

Text Mining

- *Extratores de Expressões Relevantes*
- *Análise Simbólica e análise estatística de textos*
- *Descritores de Documentos*
- *Classificação supervisionada vs não-supervisionada de Documentos*
- *Casos de estudo em Text Mining*

4.4.5. Syllabus:

Introduction

What is data? Examples of data analytic tasks and perspectives

Mining of structured or unstructured data

Data Analytics

- *Data Understanding*
- *Data Pre-processing*
- *Dimensionality Reduction*

Principal Component Analysis

Singular Value Decomposition

- *Descriptive Modeling (crisp and fuzzy)*

Representative-based clustering algorithms

Spectral clustering

- *Validation and Interpretation of Clusters*
- *Case Studies*

Text Mining

- *Extractors of Relevant Expressions*
- *Symbolic and Statistical Analysis of texts*
- *Document Descriptors*
- *Extraction of explicit and Implicit Keywords. Summarization*
- *Supervised vs unsupervised Document Classification*
- *Text Mining Case Studies*

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

São apresentadas e discutidas várias abordagens de estado da arte em data analytics e text mining de acordo com a bibliografia adotada. As várias metodologias estudadas são enquadradas com exemplos de aplicação e casos de estudo.

Os trabalhos práticos e exercícios tutoriais permitem consolidar conhecimentos na implementação/adaptação dos algoritmos estudados a problemas reais orientados aos dados.

A programação de algoritmos e de protocolos experimentais de validação orientados aos dados oferece experiência na modelação e computacional de dados.

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

Several state of the art models and methods of data analytics and text mining are presented and discussed according to the adopted text books. The methods studied are applied to practical examples and case studies.

Tutorial classes and practical assignments promote the consolidation of theoretical knowledge as well as practice in the implementation and adaptation of the algorithms to data-driven real world problems. Practical assignments also focus on the interpretation of experimental results and on the evaluation of the methods used.

The programming of algorithms and data-driven experimental protocols provide experience on computational data modeling.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas são lecionados os tópicos fundamentais da matéria, os quais deverão ser complementados com a leitura da bibliografia adotada.

As aulas laboratoriais destinam-se à orientação tutorial e realização dos trabalhos práticos. Em todas as aulas são disponibilizados os acetatos da matéria teórica.

A avaliação é composta por uma componente teórica e uma componente laboratorial.

A nota laboratorial (NL) é a média de 2 trabalhos práticos.

A frequência é concedida com $NL \geq 8.5$

Os exames são composto por 2 partes, correspondentes a cada um dos módulos.

Para efeitos de cálculo da nota teórica (NT), a nota de cada componente do exame recurso, substitui, se melhor, a classificação obtida na respetiva parte do exame de época normal.

$NT = (\max(E1_M1, E2_M1) + \max(E1_M2, E2_M2)) / 2.$

A nota final (NF) é definida pela média da nota teórica e nota laboratorial

$NF = (NT + NL)/2.$

4.4.7. Teaching methodologies (including students' assessment):

Lectures will cover the fundamental topics of the subject matter, which the students should complement with the adopted bibliography. All lecture materials will be supplied for further study.

Tutorial classes will be dedicated to exercises and guidance in the practical assignments, focusing on selected topics.

The evaluation is made by two components: theoretical and Laboratory/Project.

The Lab/Project grade (PG) is calculated as the mean of the scores of the two practical projects:

$PG = (P1 + P2) / 2.$

Access to exam requires $PG \geq 8.5$ points.

The exams of Epoc 1 and Epoc 2 are organized in two parts, each of each module.

The grade of the theoretical component (TG) is calculated by the arithmetic mean of the scores of the two parts of the exam. The theoretical grade retains the best score of each component from both exam epocs.

The final grade (FG) is calculated as the mean of the grades of the theoretical and Laboratorial/Project components:

$FG = (TG + PG) / 2.$

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

Os objetivos desta unidade curricular são:

- a) a aquisição de conhecimentos teóricos para a compreensão de várias aproximações de Data Analytics e Text Mining, das metodologias e dos princípios da sua aplicação;*
- b) a aptidão para implementar estes algoritmos e, assim, resolver problemas práticos;*
- c) a competência para avaliar a adequação de cada método a casos de estudo e para avaliar criticamente os resultados obtidos.*

O objetivo a) será atingido pela exposição da matéria, discussão e esclarecimento de dúvidas nas aulas teóricas, onde se abordará o fundamento teórico desta unidade curricular. Esta matéria será avaliada em testes individuais, onde os alunos têm um tempo limitado para mostrar o seu grau de assimilação e compreensão dos assuntos.

Os objetivos b) e c) serão atingidos principalmente pela componente prática da U.C., se bem que sempre assente nas aulas teóricas, por meio das aulas práticas e do trabalho autónomo do aluno. As aulas práticas darão aos alunos oportunidade para aplicar os conhecimentos a casos concretos, experimentar as diferentes abordagens e examinar, em detalhe, os vários algoritmos em condições realistas. A componente prática será avaliada em dois trabalhos de grupo, avaliação esta que focará não só a implementação e aplicação prática dos algoritmos mas também a capacidade crítica de avaliação dos métodos escolhidos e dos resultados. Esta componente prática contribuirá também para a autonomia dos alunos.

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

The course objectives are:

- a) the acquisition of theoretical knowledge to allow the understanding of different approximations of Data Analytics and Text Mining, methodologies and their principles of application;*
- b) the skill to implement these algorithms and, thus, solve practical problems;*
- c) the competence to evaluate the suitability of each method to case studies and for a critical evaluation of the results.*

Objective a) will be reached mainly through the theoretical lectures, which include exposition of the subject matter and discussion. These lectures will cover the theoretical foundation of the course and this knowledge will be evaluated in individual written tests, where the students will have limited time to show their success in assimilating and understanding the subjects.

Objectives b) and c) will be reached mainly through practice, during the tutorial classes and the student's own work. Tutorial classes will give the students the opportunity to apply their knowledge to specific cases, experiment with different approaches and examine, in detail, the different algorithms in realistic conditions. This practical component will be evaluated in two group assignments, focusing on the implementation and practical application of the algorithms but also on the critical assessment of the methods and results. This component will contribute to give the student the autonomy and self-reliance needed for future applications and further studies in this field

4.4.9. Bibliografia de consulta/existência obrigatória:

- Larose, D. T. , Larose C. D. (2015), Data Mining and Predictive Analytics, Wiley (2nd Edition)*
- Mirkin, B. (2019) Core Data Analysis: Summarization, Correlation, and Visualization, Springer*
- Nascimento, S. (2005). Fuzzy Clustering via Proportional Membership Model, Frontiers of Artificial Intelligence and Applications, v 119, IOS Press*
- Weiss, S.M., Indurkha, N., Zhang, T., Damerou, F. (2005), Text Mining: Predictive Methods for Analyzing*

Mapa IV - Qualidade do Software

4.4.1.1. Designação da unidade curricular:

Qualidade do Software

4.4.1.1. Title of curricular unit:

Software Quality

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Ana Maria Diniz Moreira (sem horas de contacto)

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

Miguel Carlos Pacheco Afonso Goulão -T:14; PL:14

Miguel Jorge Tavares Pessoa Monteiro -T:14; PL:14

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

Saber:

- Componentes de um sistema de gestão da qualidade (SGQ)
- Modelos de qualidade de software e sua avaliação
- Verificação e validação
- Princípios, objetivos e conceitos fundamentais da reengenharia de software
- Software legado
- Técnicas e ferramentas de engenharia reversa
- Reengenharia e refabricação de programas

Saber fazer:

- Reificar um SGQ para projectos de desenvolvimento de software
- Construir um modelo de qualidade de software
- Selecionar e utilizar técnicas de verificação e validação adequadas
- Avaliação crítica do grau de cobertura atingido com as técnicas de verificação e validação
- Analisar sistemas existentes de dimensão não trivial
- Obtenção/extração de conceitos de alto nível de sistemas legados
- Elaboração de diagnósticos úteis para a reengenharia
- Dominar técnicas de reengenharia de software

Soft skills:

- Trabalho em equipa
- Defesa pública de pontos de vista

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

Knowledge:

- Elements of a Software Quality Management (SQM) system
- Software Quality Models and evaluation
- Verification and Validation
- Principles, Concepts and Aims of Software Reengineering
- Legacy Software Systems
- Techniques and Tools for Reverse Engineering
- Program Reengineering and Refactoring

Skills and competencies:

- Reify a SQM for software development projects
- Define a software quality model
- Select and use adequate verification and validation techniques
- Critical evaluation of the coverage achieved by verification and validation techniques
- Analyze existing, non trivial, software systems
- Extract high-level information from legacy systems
- Diagnostic reporting
- Master techniques for software system reengineering

Soft skills:

- Teamwork
- Public defense of points of view

4.4.5. Conteúdos programáticos:

1. Introdução à Qualidade de Software

2. Modelos de Qualidade de Software

2.1 Modelos de Qualidade do Processo

2.2 Modelos de Qualidade do Produto

3. Verificação e Validação de Software

3.1 Revisões (passo-a-passo, inspecções, demonstrações, auditorias)

3.2 Técnicas de leitura (ad-hoc, baseada em listas de verificação, baseada em defeitos, baseada em perspectiva)

3.3 Análise de custos e benefícios das revisões

3.4 Teste de software (fundamentos, testes de caixa negra, testes de caixa branca)

4. Evolução de Software

4.1 Leis da Evolução de Software

4.2 Deterioração de Software

- 5. *Mineração de repositórios de software*
- 5.1 *Engenharia reversa de sistemas de software*
- 5.2 *Métricas de software*
- 5.3 *Visualização de software*

- 6. *Reengenharia de sistemas de software*
- 6.1 *Fundamentos de reengenharia*
- 6.2 *Refabricação*
- 6.3 *Rastreabilidade*

- 7. *Reutilização de Software*
- 7.1 *Famílias de software*
- 7.2 *Bibliotecas de componentes reutilizáveis*

4.4.5. Syllabus:

1. *Introduction to software quality*
2. *Software Quality Models and Standards*
 - 2.1 *Process Quality Models*
 - 2.2 *Product Quality Models*
3. *Software Verification and Validation*
 - 3.1 *Reviews (walkthroughs, inspections, demonstrations, audits)*
 - 3.2 *Reading techniques (ad-hoc, checklist-based, defect-based reading, perspective-based reading)*
 - 3.3 *Costs and benefits analysis of reviews*
 - 3.4 *Software testing (fundamentals, black-box testing, white-box testing)*
4. *Software evolution*
 - 4.1 *Laws of Software Evolution*
 - 4.2 *Software ageing*
5. *Mining software repositories*
 - 5.1 *Reverse engineering of software systems*
 - 5.2 *Software metrics*
 - 5.3 *Software visualization*
6. *Reengineering of software systems*
 - 6.1 *Reengineering fundamentals*
 - 6.2 *Refactoring*
 - 6.3 *Traceability*
7. *Software reuse*
 - 7.1 *Software families*
 - 7.2 *Libraries of Reusable Components*

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

O estudo da qualidade é suportado por uma visão pragmática de normas, metodologias e modelos, em linha com as boas práticas da indústria. O programa inclui uma introdução à qualidade de software (1), explorada através de abordagens de promoção da qualidade no seu desenvolvimento e evolução e ancorada em modelos de qualidade (2). Apresentam-se e praticam-se técnicas de verificação e validação (3). A prazo, para responder a novos requisitos, os sistemas têm de passar por processos de evolução no sentido de preservar ou mesmo melhorar a sua qualidade (4). Assim, um dos objetivos desta UC é ministrar abordagens e técnicas que permitam identificar problemas de evolução, usando técnicas de engenharia reversa aplicadas a casos de estudo reais (5). Uma vez identificados, estes problemas são atacados com práticas de reengenharia (6), incluindo a refabricação de software, mantendo a rastreabilidade no processo. Por fim, discute-se a reutilização, como uma técnica de promoção da qualidade (7).

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

The software quality syllabus is supported by a pragmatic view of standards, methodologies, and models, in line with industry best practices. The syllabus includes an introduction to software quality (1), aimed at promoting approaches to software quality during development and evolution, in line with quality models (2). Verification and validation techniques are covered in detail (3). To cope with new requirements, the systems need to evolve while preserving or even improving their quality (4). One of the goals of this course is to cover approaches and techniques to identify evolution problems, using reverse engineering techniques applied to real case studies (5). Once identified, these problems are tackled with reengineering practices (6) including software refactoring, while maintaining traceability throughout the process. Finally, software reuse is discussed as a quality-enhancing technique (7).

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.

Nas aulas de laboratório, os alunos resolvem exercícios que constituem exemplos concretos de aplicação dos

conceitos dados nas anteriores aulas teóricas.

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

A avaliação consiste em dois testes (num total de 45% da nota) e dois trabalhos práticos (num total de 55%) da nota. Um dos trabalhos (e respectivo teste) incide, sobretudo, em modelos de qualidade e em técnicas de verificação e validação). O outro trabalho, e respectivo teste, incide sobre a engenharia reversa, reengenharia e aplicação de padrões de desenho.

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 concretely applying the concepts discussed in the previous lectures.

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

The evaluation includes 2 tests (corresponding to a total of 45% of the final grade) and 2 projects (55% of the final grade). One of the projects, and corresponding test, focuses primarily in quality models, verification and validation. The other project, and corresponding test, focuses on reverse engineering, reengineering and the application of design patterns.

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

A apresentação das matérias nas teóricas é complementada com a exercitação de técnicas que as implementam nas aulas práticas (e nos projetos). Os alunos são expostos a boas práticas usadas nas organizações de desenvolvimento de software com um elevado grau de maturidade e têm a oportunidade de experimentar algumas delas. Os trabalhos práticos em grupo visam consolidar a utilização dessas práticas, contribuindo para o desenvolvimento de competências de trabalho em equipa e em colaboração entre diferentes equipas, em particular, no contexto de atividades de verificação e validação. Este tipo de colaboração entre equipas contribui para fortalecer aspetos tais como a coordenação inter-grupal e a defesa pública de pontos de vista, sendo ambos os aspetos instrumentais nas atividades de promoção e garantia da qualidade de software. A consolidação de conceitos que suportam as boas práticas na indústria é aferida sobretudo através dos testes de avaliação.

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

The introduction of subjects in the lectures is complemented with the usage of particular techniques that implement those subjects in the lab sessions (and in the projects). This way, students are exposed to best practices from software development organizations with a high maturity level, and are able to experiment some of those techniques. Some of the projects are carried out in groups and in cooperation with different groups, particularly in the context of verification and validation tasks. This kind of cooperation is aimed at inter-team cooperation and the public defense of points of view, both aspects being instrumental in software quality promotion and assurance. Furthermore, the consolidation of the more theoretical aspects that support these best practices in industry is assessed mostly through the evaluation tests.

4.4.9. Bibliografia de consulta/existência obrigatória:

Chikofsky, E.J., Cross II, J.H. Reverse Engineering and Design Recovery: a Taxonomy. IEEE Software 7(1), pp.13-17, IEEE Computer Society Press, January 1990.

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

Fowler M. Refactoring: Improving the Design of Existing Code. Addison-Wesley, 1999. ISBN:0-201-48567-2

Mapa IV - Requisitos e Arquitetura de Software

4.4.1.1. Designação da unidade curricular:

Requisitos e Arquitetura de Software

4.4.1.1. Title of curricular unit:

Software Requirements and 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:

168

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

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Ana Maria Diniz Moreira – T:28; 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):***RAS oferece aos alunos conhecimentos, aptidões e competências em engenharia de requisitos e arquitetura de software, facilitando a transição entre os requisitos e a arquitetura.**Compreender:*

- Fundamentos da engenharia de requisitos
- Métodos de especificação e análise de requisitos
- Fundamentos da arquitetura de software
- Padrões e estilos arquiteturais
- Derivação sistemática da arquitetura de software

Saber:

- Lidar com os desafios colocados pela dimensão, objetivos contraditórios e descrição deficiente dos problemas
- Compreender as vantagens da sistematização e reutilização planeada para a derivação da arquitetura
- Ponderar a arquitetura com base em alternativas nem sempre disjuntas

Saber Fazer:

- Aplicar métodos de engenharia de requisitos
- Refinar atributos de qualidade e estudar o impacto entre eles
- Analisar e escolher alternativas arquiteturais
- Descrever arquiteturas usando vistas, padrões e estilos
- Utilizar linguagens de descrição arquitetural e ferramentas automáticas

4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):*RAS provides students with knowledge, skills and competences in requirements engineering and software architecture, bridging the gap between requirements and architecture.**Understand:*

- Fundamentals of Requirements Engineering
- Requirements specification and analysis methods
- Fundamentals of software architecture
- Architectural standards and styles
- The base for systematic derivation of software architecture

To know:

- Dealing with challenges introduced by size, conflicting goals and poor description of the problems
- Understanding the advantages of systematization and planned reuse for architectural derivation
- Selecting architectural alternatives

Know how to do:

- Apply requirements engineering methods to identify, specify, validate and manage requirements
- Refine quality attributes and study the impact among them
- Analyze and choose architectural alternatives
- Describe architectures using views, patterns, and styles
- Use architectural description languages and automatic tools

4.4.5. Conteúdos programáticos:**1. Introdução**

- A natureza e a importância da engenharia de requisitos
- Requisitos como drivers da arquitetura de software
- Arquitetura de software e sua importância na evolução do software

2. Processo de engenharia de requisitos

- Elicitação e exploração de requisitos (*Design Thinking, Brainstorming*)
- Análise e negociação de requisitos
- Validação e gestão de requisitos
- Documentação de requisitos
- 3. Técnicas de engenharia de requisitos
- Métodos para especificação de requisitos
- Refinamento e operacionalização de requisitos não-funcionais
- Modelação baseada em objetivos
- Derivação de modelos OO
- 4. Fundamentos de desenho arquitetural
- Princípios
- Mapeamento de requisitos em conceitos arquiteturais
- Conformidade com requisitos de qualidade
- Documentação arquitetura de software
- 5. Técnicas de arquitetura de software
- Estruturação e decomposição modular
- Vistas arquiteturais
- Estilos e padrões arquiteturais
- Linguagens de descrição arquitetural
- Avaliação da arquitetura

4.4.5. Syllabus:

1. Introduction
 - The nature and importance of requirements engineering
 - Requirements as drivers of software architecture
 - Software architecture and its importance in software evolution
2. Requirements engineering process
 - Requirements elicitation and exploration (*Design Thinking, Brainstorming*)
 - Requirements analysis and negotiation
 - Requirements validation and management
 - Requirements documentation
3. Requirements engineering techniques
 - Requirements specification methods
 - Refinement and operationalization of non-functional requirements
 - Goal-based modelling
 - Derivation of object-oriented models
4. Software architecture design fundamentals
 - Principles
 - Mapping requirements to architectural concepts
 - Compliance with quality requirements
 - Documenting software architecture
5. Software architecture techniques
 - Structuring and modular decomposition
 - Architectural views
 - Architctural patterns & styles
 - Architctural description languages
 - Architecture evaluation

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

RAS estuda os vários tipos de requisitos como pilares orientadores da arquitetura de software no contexto do estado da arte atual, envolvendo por isso:

- *O estudo de técnicas próprias de engenharia de requisitos dirigida por objectivos com ênfase na especificação e análise de requisitos não-funcionais que são fundamentais para a derivação sistemática da arquitetura do software;*
- *A análise e propagação, através de mapeamentos sistemáticos, dos requisitos para a arquitetura de software com vista a facilitar a evolução do futuro sistema levando ao estudo de vistas, padrões e estilos arquiteturais, terminando com o refinamento e avaliação da arquitetura resultante.*

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

RAS studies the various types of requirements as fundamental drivers of software architecture in the context of the current state of the art. This involves:

- *The study of goal-oriented requirements engineering techniques with an intentional emphasis on the specification and analysis non-functional requirements, given their importance for the systematic derivation of an architectural design;*
- *The analysis and propagation, via systematic mappings, of the requirements to the software architecture to facilitate the evolution of the future system. This leads to the study of architectural views, patterns and styles, and finishes with the refinement and evaluation of the resulting architecture.*

4.4.7. Metodologias de ensino (avaliação incluída):

Aulas teóricas dadas em sala de aula com um projetor. Aulas práticas em laboratório equipado com computadores e projetor e têm por objetivo a aplicação das matérias lecionadas nas aulas teóricas anteriores.

A avaliação consiste num projeto com dois entregáveis e dois testes. O projeto dá a frequência na cadeira, o que obriga a uma nota mínima de 9,5 valores.

Na primeira semana de aulas os alunos recebem um caderno de encargos sobre o projeto a desenvolver ao longo do semestre (maioritariamente fora das aulas). Os dois entregáveis, avaliados separadamente, são o “documento de especificação de requisitos” (submetido aproximadamente na semana 7) e o “documento de especificação da arquitetura” (submetido na semana 14).

A nota final é uma média dos quatro elementos de avaliação, cada um com um peso de 25%.

Acesso a exame de recurso é reservado a alunos com frequência, mas com uma média final negativa, ou a alunos que desejam subir a média final.

4.4.7. Teaching methodologies (including students' assessment):

Lectures in a classroom with a projector. Labs in room with computers and projector. These sessions test the topics taught in previous theoretical lectures.

The evaluation consists of: a project with two deliverables and two evaluation tests. The project gives the “frequency”, requiring a minimum grade of 9.5 values (out of 20).

In the first week of the semester, the students receive a description of the project to be developed during the semester (mainly outside the classroom). The two deliverables, evaluated separately, are the SRS (software requirements specification) (submitted approximately in week 7) and the “architecture specification document” (submitted in week 14).

The final grade is the average of the four assessment elements, each weighing 25%.

Access to the re-sit exam is conditional upon satisfactory performance in the project (minimum grade achieved) but with a negative final average, or to students with satisfactory performance but who wish to improve their final grade.

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

As matérias apresentadas nas aulas teóricas são ilustrados nas práticas com exemplos inspirados em situações reais, garantindo a consolidação dos vários tópicos. O trabalho prático, de dimensão significativa, permite ao aluno experimentar, de forma independente e em concreto, as técnicas ensinadas desde a elicitação de requisitos até ao desenho da arquitetura e sua avaliação. Os trabalhos salientarão a importância dos requisitos não-funcionais e seus tradeoffs, levando o aluno compreender a necessidade de usar estruturas standard (como padrões, estilos) para o desenho da arquitetura.

Neste contexto, os alunos compreenderão ainda as vantagens da reutilização planeada. O projeto endereça problemas atuais da sociedade, desafiando a curiosidade e vontade do aluno em ir para além do que é ensinado nas aulas. Os milestones do trabalho prático são discutidos e verificados na aula prática. O trabalho prático fomenta o trabalho em equipa, e os testes avaliam o desempenho individual.

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

The topics presented in lectures are illustrated in the lab with examples inspired by real situations, ensuring the consolidation of the various topics. A project, of a significant size, provides an opportunity for students to experiment the techniques taught, from requirements elicitation to the design of software architecture and its evaluation.

The examples stress the importance of non-functional requirements and their required tradeoffs, allowing students to understand the need for standard architectural structures (such as patterns, styles). In this context, students should see the benefits of planned reuse. The project typically addresses current problems, challenging the students' curiosity and willingness to go beyond what is taught in class. The project milestones are discussed and assessed in practical classes, so that the work progress is ensured. The project fosters teamwork, while the tests assess individual performance.

4.4.9. Bibliografia de consulta/existência obrigatória:

- K. Wiegers, J. Beatty, *Software Requirements*, Microsoft Press, 2013
- A. Lamsweerde, *Requirements Engineering*, Wiley, 2009
- I. Alexander, N. Maiden, *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*, Wiley, 2004
- I. Alexander, R. Stevens, “*Writing Better Requirements*”, Addison Wesley, 2002
- R. N. Taylor, et al, “*Software Architecture: Foundations, Theory, and Practice*” John Wiley and Sons, 2009
- Mary Shaw and David Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.
- L. Bass, P. Clements, R. Kazman, “*Software Architecture in Practice*”, 2nd edition, Addison-Wesley, 2003
- P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, J. Stafford, “*Documenting Software Architectures: Views and Beyond*”, Addison-Wesley, 2003.
- M. Fowler, D. Rice, M. Foemmel, E. Hieatt, R. Mee, R. Stafford, *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2002.

Mapa IV - Segurança de Software

4.4.1.1. Designação da unidade curricular:

Segurança de Software

4.4.1.1. Title of curricular unit:

Software Security

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:28; PL:28***4.4.1.6. ECTS:**

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Luís Manuel Marques da Costa Caires - T:28; 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):***Nesta unidade curricular de especialização, os estudantes aprendem princípios, métodos e técnicas importantes para reforçar a segurança do software durante o desenvolvimento e a construção de aplicações. Ele complementa e aprofunda outras UCs de do curso que cobrem aspetos de segurança mas que se concentram em aspectos da segurança no nível de sistema ou rede.***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***In this course students learn key principles, methods, and techniques for enforcing security in software during development and construction. It complements but is very different in scope from other security courses that focus on different aspects of security at the system or network communication level.***4.4.5. Conteúdos programáticos:**

- 1. Conceitos gerais de segurança de software. Definição e modelação de ameaças e atacantes. Propriedades de segurança como invariantes do sistema.*
- 2. Princípios de design de software seguro. Preservação de segurança entre os módulos e confiabilidade.*
- 3. Autenticação. Modelos e objetivos da autenticação. Técnicas básicas de autenticação em aplicações Web. Mecanismos de delegação.*
- 4. Autorização. Políticas e regras de controle de acesso. Linguagens para expressar e aplicar autorização. Permissões e capacidades.*
- 5. Fluxos de Informação. Confidencialidade e Integridade. Não interferência. Segurança do fluxo de informações com base em linguagens.*
- 6. Ameaças de segurança em Aplicações Web e linguagens inseguras. Contramedidas para ameaças.*
- 7. Segurança e proveniência de dados. Segurança em bases de dados relacionais. Controle de acesso em modelos de dados. Modelos e linguagens para expressar a proveniência.*

4.4.5. Syllabus:

- 1. Software Security Concepts. Attacker Models. Security properties and policies. Security Properties as System Invariants.*
- 2. Basic principles: Least Privilege; Fail-Safe Defaults; Economy of Mechanism;; Separation of Duties; Least Common Mechanism.*
- 3. Authorization and access control models. Policies and rules. Stack inspection, signed code. Permissions capability models.*
- 4. Information Flow. Security Lattices. Non-interference. Declassification. Covert Channels and indirect flows. (Sand)boxing and Tainting. Ref: Data Flow and Type-based analysis..*
- 5. Threats. Web Applications (code injection, cross-site scripting, cross-site forgery, session hijacking). Unsafe Languages (buffer overruns, stack smashing). Countermeasures (information flow, capabilities, tainting, monitors).*
- 6. Data Security and Provenance. Privacy and utility; statistical database security; differential privacy. Provenance models and languages.*

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

Nesta unidade curricular, os estudantes aprendem os princípios, métodos e técnicas importantes para reforçar a segurança do software durante o desenvolvimento e a construção de aplicações. Ele complementa, mas tem escopo muito diferente de outros cursos de segurança que se concentram em diferentes aspectos da segurança no nível de comunicação do sistema ou da rede.

Os princípios são abordados nas seções 1 - 5, os métodos e técnicas cobertos nas seções 5-7. Estes são abordados de maneira coerente e apresentam os principais conceitos de confidencialidade, integridade e responsabilidade. Os principais casos de uso são abordados na seção 6. No projeto de laboratório, os alunos desenvolvem o núcleo de uma estrutura básica de segurança aplicacional a utilizam-na para promover a segurança de uma aplicação web minimalista.

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

In this course students learn key principles, methods, and techniques for enforcing security in software during development and construction. It complements but is very different in scope from other security courses that focus on different aspects of security at the system or network communication level. Principles are covered in sections 1 and 2 and 6. methods and techniques are covered in sections 3 and 4. These are covered in a coherent way, around the key concepts of confidentiality, integrity and accountability. Key use cases are covered in section 5. In the lab project students develop the core of a basic application security framework, and use it to enforce security of a toy web application.

4.4.7. Metodologias de ensino (avaliação incluída):

Muitas falhas de segurança ocorrem muitas vezes devido a erros de design ou programação no nível das aplicações. Esta unidade curricular cobre princípios, métodos e técnicas para reforçar a segurança no software, com ênfase no nível das aplicações e cobrindo aspectos relacionados tanto com o código como com os dados. Os estudantes desenvolvem também um projeto supervisionado de uma aplicação web centrada em dados completo e totalmente segura, de acordo com um conjunto bem definido de requisitos gerais e políticas específicas do domínio da aplicação. Como ponto de partida, é fornecido aos estudantes o código fonte de uma aplicação simples bem estruturada, para viabilizar a focalização do ensino nos aspetos relevantes à unidade curricular.

Avaliação: dois testes (35% cada); dois trabalhos (15% cada)

4.4.7. Teaching methodologies (including students' assessment):

Many security flaws, such as insecure dataflows or leaks, are also often due to design or programming errors at the application stack level. The key to defending against application-level security breaches requires considering application-level security. This course focus on principles, methods, and techniques for enforcing security in software, with an emphasis on the application level, and covering aspects related to both code and data. Students also develop a supervised project of a complete data-centric application fully secure according to a well-defined set of general requirements and specific application level policies. The skeleton and source code of a well-designed application will be given to the students as a starting point.

Assessment: two tests (midterm and final, 35% each); two programming assignments (15% each)

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

Embora a segurança nos sistemas computacionais possa ser em grande parte reforçada ao nível da pilha do sistema operacional, a segurança a nível sistema concentra-se na mitigação de ataques de baixo nível (políticas de controle de acesso, proteção de repositórios de dados específicos ou protocolos de comunicação criptográfica), mas muitas violações de segurança surgem ao nível das aplicações. Muitas falhas de segurança, como fluxos de dados inseguros, também são frequentemente causadas por erros de design ou de programação no nível das aplicações. A ênfase da UC é pois nos aspetos relacionados com a construção de aplicações seguras, e cobrem aspectos relacionados tanto com o código como com os dados. Durante a UC, os estudantes desenvolvem, de forma supervisionada, o projeto de uma aplicação completa, centrado em dados e totalmente segura de acordo com um conjunto bem definido de requisitos gerais e políticas específicas, expressas ao nível da aplicação. O esqueleto e o código fonte de uma aplicação bem projetada é fornecida aos estudantes, como ponto de partida.

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

Although security in computing systems can be largely enforced at the level of operating system stack, operating-system security focus on low-level attacks (such as access control policies, protecting particular files, or cryptographic communication protocols), many attacks are high-level, or application-level (such as email worms that pass by access controls pretending to be executed on behalf of a mailer application). Many security flaws, such as insecure dataflows or leaks, are also often due to design or programming errors at the application stack level. The key to defending against application-level security breaches requires considering application-level security. The emphasis of the course is thus on the application level, and covering aspects related to both code and data. During the course, students develop a supervised project of a complete data-centric application fully secure according to a well-defined set of general requirements and specific application level policies. The skeleton and source code of a well-designed application will be given to the students as a starting point.

4.4.9. Bibliografia de consulta/existência obrigatória:

Secure Software Lifecycle, CyBok chapter by Laurie Williams, 2019
Software Security, CyBok chapter by Frank Piessens, 2018
Avoiding the Top 10 Security Design Flaws, IEEE, 2014.

*The Tangled Web: A Guide to Securing Modern Web Applications, Michał Zalewski, 2011
Papers and web links.*

Mapa IV - Sistemas de Computação Móvel e Ubíqua

4.4.1.1. Designação da unidade curricular:

Sistemas de Computação Móvel e Ubíqua

4.4.1.1. Title of curricular unit:

Mobile and Pervasive Computing

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:28; PL:28

4.4.1.6. ECTS:

6

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Nuno Manuel Ribeiro Preguiça (sem horas de contacto)

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

Carmen Pires Morgado - T:14; PL:14

Maria Cecília Farias Lorga Gomes - T:14; PL:14

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

Esta é uma unidade curricular (UC) opcional sobre sistemas de computação móvel e ubíqua. Introduce os fundamentos dos ambientes de computação móvel e ubíqua, além dos modelos e técnicas para o desenvolvimento de aplicações para estes ambientes.

Como pré-requisito, é necessário possuir conhecimentos de programação e sistemas distribuídos.

Saber:

+ *Características dos ambientes de computação móvel e ubíqua*

+ *Arquiteturas, modelos e técnicas de programação para estes ambientes*

+ *Conhecimentos fundamentais de redes móveis*

+ *Técnicas de localização*

+ *Problemas e soluções específicas para aplicações móveis (interfaces, gestão de dados e adaptação ao contexto)*

+ *Problemas e técnicas de sistemas de sensoriamento*

Saber Fazer:

+ *Desenho e realização de sistemas móveis*

+ *Implementação de soluções para lidar com aspetos específicos relacionados com a mobilidade*

+ *Desenho e implementação de sistemas ubíquos*

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

This is an optional course on mobile and pervasive computing systems. It provides the basic knowledge on the characteristics of mobile and pervasive computing, models and techniques used in the development of applications targeting these environments. As prerequisites, students should have previous acquaintance with programming and distributed systems.

Knowledge:

+ *Characteristics of mobile and pervasive computing*

+ *Architectures, models and techniques for programming mobile and pervasive computing systems*

- + *Basic knowledge on mobile networks*
- + *Location techniques*
- + *Specific problems and solutions for mobile applications (interfaces, data management, context awareness)*
- + *Problems and techniques for sensing systems*
- Application:
 - + *Design and implement mobile systems*
 - + *Implement solutions to address mobility specific issues*
 - + *Design and implement pervasive systems*

4.4.5. Conteúdos programáticos:

1. *Introdução, exemplos e desafios*
2. *Introdução à computação ubíqua*
 1. *Dispositivos*
 2. *Modelos e arquiteturas*
 3. *Sistemas de sensoriamento*
 1. *Redes de sensores*
 2. *IoT systems*
 3. *Sensoriamento participativo*
 4. *Localização*
 1. *Conceitos de localização*
 2. *Técnicas de localização*
 5. *Programação de sistemas móveis*
 1. *Modelos e arquiteturas*
 2. *Frameworks nativos*
 3. *Soluções baseadas na web*
 6. *Redes sem fios*
 1. *Redes sem fios*
 2. *Mobilidade*
 7. *Aspetos específicos da computação móvel*
 1. *Interfaces*
 2. *Gestão de dados*
 3. *Adaptação ao contexto*

4.4.5. Syllabus:

1. *Introduction, examples, and challenges.*
2. *Introduction to pervasive computing*
 1. *Devices*
 2. *Models for pervasive computing*
3. *Pervasive Sensing*
 1. *Sensor networks*
 2. *IoT systems*
3. *Participatory sensing*
4. *Mobile Computing Specific Issues*
 1. *Interfaces*
 2. *Data management*
 3. *Context-awareness*
5. *Programming mobile systems*
 1. *Models and architectures*
 2. *Native frameworks*
 3. *Web-based solutions*
6. *Wireless networks*
 1. *Wireless networks*
 2. *Mobility*
7. *Mobile Computing Specific Issues*
 1. *Interfaces*
 2. *Data management*
 3. *Context-awareness*

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 lhe permitam desenhar e desenvolver aplicações em sistemas de computação móvel e ubíqua.

No contexto da computação móvel, apresenta as aproximações gerais de programação de aplicações nestes ambientes, seguida de aspetos específicos dos ambientes móveis, incluindo redes sem fios, sistemas de localização, interfaces, gestão de dados e adaptação ao contexto.

No contexto da computação ubíqua, estudam-se os aspetos relacionados com a programação e os sistemas de sensoriamento.

Na componente prática, os conceitos estudados na parte teórica serão colocados em prática em aplicações móveis, permitindo aos estudantes ganhar experiência no desenvolvimento deste tipo de sistemas.

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 applications for mobile and pervasive computing environments. To reach this goal, the course includes two parts focusing on mobile computing and pervasive computing respectively.

In the context of mobile computing, it starts by addressing the general solutions for programming mobile systems, and continues by addressing mobile systems issues, including wireless networks, location techniques, interfaces, data management and context adaptation.

In the context of pervasive systems, the course will focus on programming approaches and sensing systems.

In the labs, the topics studied in the lectures will be exercised in the development of mobile systems focusing on different aspects, allowing students to gain experience in programming this type of systems.

4.4.7. Metodologias de ensino (avaliação incluída):

Componente teórica com exposição da matéria e realização de um projeto integrado nas aulas de laboratório ao longo do semestre, através do qual o aluno consolida os conhecimentos adquiridos.

A avaliação tem duas componentes, uma teórica (com um peso de 2/3 na nota final) e uma prática (com um peso de 1/3).

A avaliação da componente teórica consiste em dois testes individuais ou exame.

A avaliação da componente prática consiste num projeto de programação em grupo.

4.4.7. Teaching methodologies (including students' assessment):

A theoretical component with weekly classes focusing on introducing and discussing the subjects. A practical component in the form of an integrated project realized weekly in the laboratory classes, with the goal of consolidating the acquired knowledge.

The assessment has two components, one theoretical (with a weight of 2/3 in the final grade) and a practical (with a weight of 1/3).

The theoretical evaluation consists of two individual tests or an examination.

The laboratory evaluation consists of a programming project in a group of students.

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

A UC visa fornecer um conjunto de conhecimentos e competências que permitam ao aluno desenhar e desenvolver aplicações em sistemas de computação móvel e ubíqua. Para atingir esse objetivo, a UC aborda os vários temas da computação ubíqua e os problemas relacionados com a computação móvel.

No contexto da computação ubíqua, estudam-se os aspetos relacionados com a programação e os sistemas de sensoriamento e localização.

No contexto da computação móvel, apresenta as aproximações gerais de programação de aplicações nestes ambientes, seguida de aspetos específicos dos ambientes móveis, incluindo redes sem fios, sistemas de Internet of Things, gestão de dados e adaptação ao contexto.

Na componente prática, os conceitos estudados na parte teórica serão colocados em prática através de um projeto integrador das diferentes áreas estudadas, em que os alunos consolidam os conhecimentos adquiridos e ganham experiência no desenvolvimento de um protótipo funcional.

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

This course aims to provide students with a set of knowledge and know-how that allows them to develop applications for mobile and pervasive computing environments. To reach this goal, the course includes two parts focusing on mobile computing and pervasive computing respectively.

In the context of pervasive systems, the course will focus on programming approaches and sensing and location systems.

In the context of mobile computing, it starts by addressing the general solutions for programming mobile systems, and continues by addressing mobile systems issues, including wireless network and Internet of Things systems, data management and context adaptation.

In the labs, the topics studied in the lectures will be exercised in the development of a mobile system project, focusing on different aspects of mobile pervasive computing studied, allowing students to gain experience in programming this type of systems.

4.4.9. Bibliografia de consulta/existência obrigatória:

The syllabus contents are presented throughout the semester through a set of slides that highlight the main concepts, as well as a set of articles updated annually, which translate the state of the art in the different domains, including the latest solutions and technologies.

Some bibliography:

- *Fundamentals of Wireless Sensor Networks: Theory and Practice (Wireless Communications and Mobile Computing), Walteneus Dargie, Christian Poellabauer, 2010.*
- *Computer Networks: A Top-Down Approach, J. Kurose, K. Ross, Addison-Wesley, 6th edition, 2013.*

Mapa IV - Teoria de Jogos Computacional

4.4.1.1. Designação da unidade curricular:

Teoria de Jogos Computacional

4.4.1.1. Title of curricular unit:

*Computational Game Theory***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:28; PL:28***4.4.1.6. ECTS:**

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***João Alexandre Carvalho Pinheiro Leite - T:28;***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Jorge Carlos Ferreira Rodrigues da Cruz - PL:28***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Esta UC irá abranger os conceitos básicos e computacionais da Teoria dos Jogos (TJ) e Desenho de Mecanismos. Os objetivos são:*

- *Introduzir a noção de jogo, os seus variados conceitos de solução*
- *Formalizar a noção de pensamento estratégico e escolha racional, e fornecer conhecimento sobre a adoção da TJ no desenho de aplicações*
- *Estabelecer as ligações entre a TJ e Informática com ênfase nos aspectos computacionais*

Conhecimento:

- *A noção de um jogo estratégico, seus equilíbrios, e as características das principais aplicações desses conceitos*

Aptidões:

- *Analisar os jogos estratégicos e calcular os seus equilíbrios.*
- *Dada uma situação de mundo real, identificar os seus aspectos estratégicos chave e, com base nestes, ser capaz de liga-los aos conceitos da TJ*

Soft-Skills:

- *Aplicação de conhecimentos teóricos formais em aplicações práticas*
- *Estar ciente de resultados de impossibilidade e trade-offs no desenho*
- *Compreender as principais ligações e interações entre TJ e Informática*

4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):*This course will cover the basic and computational concepts of Game Theory (GT) and Mechanism Design. The aims are:*

- *To introduce the notion of game, its various solution concepts*
- *To formalise the notion of strategic thinking and rational choice, and to provide insights into adopting GT in modelling applications*
- *To establish the links between GT and CS, with an emphasis on the computation aspects*

Knowledge:

- *Understand the notion of a strategic game and equilibria, and understand the characteristics of the main applications of these concepts*

Know-how:

- *Understand how to analyse strategic games and compute equilibria*
- *Given a real world situation be able to identify its key strategic aspects and, based on these, be able to connect them to appropriate game theoretic concepts*

Soft-Skills:

- *Apply formal theoretical knowledge in real-world applications*
- *Become aware of impossibility results and design trade-offs*
- *Understand the key connections and interactions between GT and CS*

4.4.5. Conteúdos programáticos:*Teoria dos Jogos - Parte I*

1. *Introdução à Teoria dos Jogos não-cooperativos: Jogos na forma normal*
2. *Computação de soluções de jogos na forma normal*
3. *Jogos com ações sequenciais: Raciocínio e Computação em jogos na forma extensiva*
4. *Representações mais ricas: para além das formas normal e extensiva*
5. *Teoria de Jogos de Coligação*

Parte II – Desenho de Mecanismos

6. *Agregando Preferências: Escolha Social*
7. *Protocolos para jogadores estratégicos: desenho de mecanismos*
8. *Protocolos para Alocação de Recursos: Leilões*

Estudos de Caso Ilustrativos (exemplos)

- *Leilões de pesquisa patrocinada da Google*
- *Preços de reserva em leilões de palavras-chave da Yahoo!*
- *Leilões do eBay*
- *Leilões de largura de banda para comunicações sem fio*
- *Troca de rins. Emparelhamentos estáveis.*
- *Mercados de negociação de eletricidade*
- *Excesso de provisionamento de redes*

4.4.5. Syllabus:*Part I - Game Theory*

1. *Introduction to Noncooperative Game Theory: Games in Normal Form*
2. *Computing Solution Concepts of Normal-Form Games*
3. *Games with Sequential Actions: Reasoning and Computing with the Extensive Form*
4. *Richer Representations: Beyond the Normal and Extensive Forms*
5. *Coalitional Game Theory*

Part II - Mechanism Design

6. *Aggregating Preferences: Social Choice*
7. *Protocols for Strategic Players: Mechanism Design*
8. *Protocols for Resource Allocation: Auctions*

Illustrative Case Studies (examples)

- *Google's sponsored search auctions*
- *Reserve prices in Yahoo! keyword auctions*
- *eBay auctions*
- *Wireless spectrum auctions*
- *Kidney exchange. Stable matching.*
- *Electricity trading markets*
- *Network over-provisioning*

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

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

As aptidões objetivo da unidade curricular decorrem da exercitação do programa em atividades práticas laboratoriais. Todo o conteúdo programático contribui para o objetivo relacionado com a capacidade de identificação de aspetos estratégicos em situações reais e sua ligação a conceitos teóricos.

Todas as competências são também fomentadas pela forma de avaliação da unidade curricular, quer pelos testes/exame quer pelas implementações desenvolvidas, e pela participação no torneio.

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

The syllabus closely follows the established knowledge objectives, thus being clear the coherence between them.

The know-how goals of the curricular unit follow from the practical exercises and implementations developed in practical laboratory activities.

All items in the syllabus contribute to the objective related to the ability to identify strategic aspects in real situations and their link to theoretical concepts.

All competencies are also fostered by the curricular unit's evaluation method, both by the tests / exams, by the implementations, and by the participation in the tournament.

4.4.7. Metodologias de ensino (avaliação incluída):

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

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

A avaliação de conhecimentos inclui:

- 2 testes individuais teóricos (ou um exame de recurso), onde é avaliado o conhecimento que os alunos adquiriram dos conceitos e características das técnicas leccionadas (70%) (nota mínima na componente: 9,5 val.).*
- Vários mini-trabalhos práticos, elaborados em grupos de 2 estudantes, que culminam com o desenvolvimento e implementação de vários jogadores autónomos para participarem num torneio que decorre no final do semestre (30%)*

4.4.7. Teaching methodologies (including students' assessment):

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

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

The evaluation includes:

- 2 individual tests (or an exam) where the knowledge acquired by the students on the concepts and techniques is assessed (70%) (minimum grade in component: 9.5 val).*
- Several mini practical projects, developed in groups of 2 students, which lead to the development and implementation of several autonomous players to participate in a tournament that takes place at the end of the semester (30%).*

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

Os objetivos da unidade curricular identificam a) a aquisição de conhecimento sobre teoria de jogos e desenho de mecanismos; b) a aplicação prática dos conhecimentos adquiridos.

O objetivo a) é fundamentalmente atingido nas aulas teóricas com a exposição de um conjunto de conceitos relacionados com o tema da cadeira, sempre acompanhados pela apresentação de casos concretos que ilustrem a sua relevância.

O objetivo b) é fundamentalmente atingido nas aulas práticas através da implementação dos algoritmos de teoria de jogos que implementam os vários conceitos de solução, quer através do desenho de jogadores autónomos.

As matérias relativas ao objetivo a) são avaliadas nos testes/exame individuais. As matérias relativas ao objetivo b) são avaliadas nos trabalhos práticos e na participação no torneio.

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

The objectives of this course include a) the acquisition of knowledge about game theory and mechanism design; b) the practical application of the acquired knowledge.

Objective a) is fundamentally achieved in lectures with the presentation of a set of concepts related to the subject at hand, always accompanied by the presentation of concrete use-cases that illustrate their relevance.

Objective b) is fundamentally achieved in practical classes through the implementation of game theory algorithms that implement the various solution concepts, and also through the design of autonomous players.

Objective a) is essentially assessed in the individual tests/exam. Objective b) are assessed in the practical work and participation in the tournament.

4.4.9. Bibliografia de consulta/existência obrigatória:

Livros de Texto / Text Books

- Noam Nisan, Tim Roughgarden, Eva Tardos and Vijay V. Vazirani (Eds.), Algorithmic Game Theory, Cambridge University Press, 2007.*
- Yoav Shoham and Kevin Leyton-Brown, Essentials of Game Theory: A Concise Multidisciplinary Introduction, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2008.*
- Yoav Shoham and Kevin Leyton-Brown, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Cambridge University Press, 2009.*
- Michael Maschler, Eilon Solan and Shumel Zamir, Game Theory, Cambridge University Press, 2013.*

Mapa IV - Visualização e Análise de Dados

4.4.1.1. Designação da unidade curricular:

Visualização e Análise de Dados

4.4.1.1. Title of curricular unit:

Visualization and Data Analytics

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:

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

6

4.4.1.7. Observações:*Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***João Carlos Gomes Moura Pires – T:28; 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):***Saber*

- *O que é Visualização Interativa de Dados (VID).*
- *O papel da interação na VID.*
- *O papel da VID na Análise Exploratória de Dados (AD) e no desenho de modelos de aprendizagem automática.*
- *O conceito de Variável Visual.*
- *Técnicas de visualização e análise de dados para dados multivariados, dados geo-espaciais e dados dependentes do tempo.*
- *Os principais componentes e características gerais dos sistemas de Visualização Analítica (VA).*
- *Avaliação de técnicas VID e sistemas de VA.*

Fazer

- *Escolher as técnicas de visualização mais apropriadas a um conjunto de dados e de objetivos.*
- *Usar um sistema VID para explorar e visualizar conjuntos de dados.*
- *Desenhar e implementar uma solução VAD para um conjunto de objetivos de analíticos.*

Soft-Skills

- *Perceber a natureza multidisciplinar desta área e entender o seu relacionamento com outras áreas do conhecimento e da engenharia.*
- *Explorar a natureza experimental do desenho de sistemas de VA.*

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

- *What is Interactive Data Visualization (IDV).*
- *The role of interaction in IDV.*
- *VID's role in Exploratory Data Analysis (DA) and the design of machine learning models.*
- *The concept of Visual Variable.*
- *Visualization and data analysis techniques for multivariate data, spatial data and time dependent data.*
- *The main general components of Analytical Data Visualization (ADV) systems.*
- *Methodologies for comparison and evaluation of ADV techniques and systems.*

Do

- *Choose the visual variables and visualization techniques most appropriate to a data set and objectives.*
- *Use a IDV system to explore and view one or more datasets.*
- *Design and implement a ADV solution for a data class and for a set of exploration objectives.*
- *Experimental bench to evaluate a IDV technique.*

Soft-Skills

- *Understand the multidisciplinary nature of this area and understand its relationship with other areas of knowledge and engineering.*
- *Explore the experimental nature of ADV system design.*

4.4.5. Conteúdos programáticos:*Introdução à visualização de dados*

- *O que é visualização?*
- *Relação entre Visualização e Outros Campos.*
- *O processo de visualização.*
- *O papel da Visualização Interativa de Dados na Análise Avançada de Dados.*
- *Fundamentos de dados.*
- *Percepção humana e processamento de informações.*

- *Semiologia dos símbolos gráficos.*
- *As variáveis visuais.*
- Técnicas de visualização para:*
 - *Dados espaciais*
 - *Dados geoespaciais*
 - *Dados orientados a tempo*
 - *Dados multivariados*
 - *Árvores, gráficos e redes*
- Conceitos e Técnicas de Interação*
 - *Operadores, operandos e espaços de interação*
 - *Componentes dos sistemas de análise visual*
 - *Projetando visualizações eficazes*
 - *Comparando e avaliando técnicas de visualização*
- Sistemas de visualização de dados analíticos*
 - *O papel do VID na análise exploratória de dados*
 - *O papel do VID nos modelos de dados explicativos*
 - *Sistemas modernos de visualização de dados analíticos*
- Linhas de investigação em visualização de dados analíticos*

4.4.5. Syllabus:

- Introduction to Data Visualization*
 - *What Is Visualization?*
 - *Relationship between Visualization and Other Fields.*
 - *The Visualization Process.*
 - *The role of Interactive DV in the Advanced Data Analytics*
 - *Data Foundations.*
 - *Human Perception and Information Processing.*
 - *Semiology of Graphical Symbols.*
 - *The Visual Variables.*
- Visualization Techniques for:*
 - *Spatial Data*
 - *Geospatial Data*
 - *Time-Oriented Data*
 - *Multivariate Data*
 - *Trees, Graphs, and Networks*
- Interaction Concepts and Techniques*
 - *Interaction Operators, Operands and Spaces*
 - *Components of Visual Analytics Systems*
 - *Designing Effective Visualizations*
 - *Comparing and Evaluating Visualization Techniques-*
- Analytical Data Visualization Systems*
 - *The role of IDV in exploratory Data Analysis*
 - *The role of IDV in Explanatory Data Models*
 - *Modern Analytical Data Visualization Systems*
- Research Directions in Analytical Data Visualization*

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

O programa inclui e assenta na compreensão dos fundamentos da Visualização de Dados e nas componentes principais dos sistema de Visualização Interativa de Dados e o seu papel central nos soluções de visualização analítica.

Os alunos tomam contato com um leque variado de técnicas de Visualização de Dados, acompanhado de uma discussão crítica da sua adequação aos dados e objetivos. Algumas destas técnicas são trabalhadas com um nível de profundidade recorrendo a dados reais e sistemas reais permitindo desenvolver as bases para saber aplicar em cenários reais.

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

The program includes and builds on understanding the fundamentals of Data Visualization and the key components of Interactive Data Visualization systems and their central role in visual analytical solutions.

Students come into contact with a wide range of Data Visualization techniques, accompanied by a critical discussion of their suitability for the data and analytical requirements. Some of these techniques are worked with a deep level using real data and real systems, allowing to develop the bases to know how to apply in real scenarios.

4.4.7. Metodologias de ensino (avaliação incluída):

Nas aulas teóricas é apresentada a matéria, com exemplos e referências a sistemas relacionados. As aulas laboratoriais destinam-se à realização do projecto que acompanha a matéria apresentada nas aulas teóricas. A avaliação do curso consiste em:

um projeto, com várias fases (especificação, estado da arte e código / interface) que, juntos, representam um projeto para desenvolver uma aplicação de visualização interactiva de dados, usando dados reais.

- um exame, se possível, caso contrário, um exame oral (remoto)

Nota final:

= 40% (oral ou exame) + 25% Artigo +35% Implementação e discussão do projecto

A aprovação à disciplina exige as seguintes notas:

(Oral ou Exame) >= 10 E

(média (Artigo; Implementação) >= 10

4.4.7. Teaching methodologies (including students' assessment):

In the lectures the course content is presented, illustrated with application examples and references to related work. The laboratory classes are intended for specification, development and presentation of the project that deals with topics presented during the lectures.

The evaluation of the course consists of:

- one project, with several phases (specification, state of the art and code/interface) that together account for a project to develop a interactive data visualization application, using real data.
- an exam if possible, otherwise an oral (remote)

Final grade formula:

Final_Grade = 40% (Oral or Exam) 25% Paper + 35% Implementation and Project discussion

Course approval requires the following minimal grades:

Oral or Exam) >= 10) AND

(mean (Paper; Implementation) >= 10)

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

A articulação entre a teoria e a prática é absolutamente essencial nesta área. O sucesso desta articulação assenta: (i) as aulas teóricas onde os fundamentos são apresentados e discutidos com os alunos e as principais técnicas de visualização e interação são expostas a um razoável nível de profundidade; as aulas práticas em que em parte delas os alunos desenvolvem competências de visualização e sobretudo visualização analítica usando as mais modernas ferramentas (ex. Tableau) e consolidam os conhecimentos teóricos; o projeto que sendo desenvolvido por fases se aplica a dados e questões analíticas do interesse de cada grupo de alunos; (iii) o projeto requer ainda a escrita de um artigo e a apresentação do trabalho oralmente.

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

The articulation between theory and practice is absolutely essential in this area. The success of this articulation rests on: (i) the theoretical lectures where the fundamentals are presented and discussed with the students and the main visualization and interaction techniques are exposed to a reasonable level of depth; practical labs in which students develop visualization skills and above all analytical visualization using the latest tools (eg Tableau) and consolidate theoretical knowledge; the project being developed in phases applies to data and analytical issues of interest to each group of students; the project also requires the writing of an article and the presentation of the paper orally.

4.4.9. Bibliografia de consulta/existência obrigatória:

- *Interactive Data Visualization: Foundations, Techniques, and Applications, Second Edition. Matthew O. Ward, Georges Grinstein, Daniel Keim, 2015, ISBN 9781482257373*
- *Visualization Analysis & Design, Tamara Munzner, 2015, ISBN: 9781466508910 ISBN (e-Book): 9781498707763*
- *Artigos actuais e a propósito de conjuntos de dados ou técnicas que sejam usados nas práticas ou nos projetos (papers about data sets or techniques that are used in practices or projects)*

Mapa IV - Web Geográfica

4.4.1.1. Designação da unidade curricular:

Web Geográfica

4.4.1.1. Title of curricular unit:

GeoWeb

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:28; PL:28***4.4.1.6. ECTS:***6***4.4.1.7. Observações:***Optativa***4.4.1.7. Observations:***Optional***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Nuno Correia (sem horas de contacto)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Maria Armada Simenta Rodrigues Grueau – T:28; PL:28***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Saber:*

- 1. Conhecer os vários elementos que compõem a Web Geográfica*
- 2. Como se representa computacionalmente o espaço em sistemas de base geográfica*
- 3. Fornecer as várias ferramentas e métodos para o desenvolvimento de sistemas na Web Geográfica*
- 4. Conhecer o ciclo de desenvolvimento de uma aplicação Web de base geográfica*

Saber Fazer:

- 5. Utilizar as potencialidades de várias tecnologias associadas à Web Geográfica*
- 6. Desenvolver um projeto Web SIG de pequena dimensão*

Competências Complementares:

- 7. Explorar capacidades de gestão de tempo e organização*
- 8. Trabalho de equipa*
- 9. Apresentação escrita e oral de trabalhos desenvolvidos*
- 10. Saber reconhecer e avaliar as necessidades de utilizadores*

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

- 1. To Know the elements which compose the GeoWeb*
- 2. To know how to computationally represent space in Geographically-based systems*
- 3. To provide tools and methods for the development of systems for the GeoWeb*
- 4. To know the development cycle of applications for the GeoWeb*

Application:

- 5. To use the potential of GeoWeb Technology*
- 6. To develop a small dimension Web GIS project*

Soft-skills:

- 7. Time management and organization*
- 8. To present work in oral and written form*
- 9. To recognize and evaluate users' needs*

4.4.5. Conteúdos programáticos:

- A. Introdução à Informação Geográfica: Natureza e funcionalidade da IG*
- B. Representação de Informação Geográfica: Modelos baseados em propriedades; Modelos baseados em entidades; Estruturas de representação*
- C. Georeferenciação: Sistema de coordenadas e de projeção (exemplos)*
- D. A Web Geográfica: Conceito genérico, Capacidades e Visão de futuro*
- E. Distribuição de Dados na Web Geográfica: Normalização de dados geográficos; Metadados para objetos geográficos; Bases de Dados Geográficos*
- F. Distribuição de Processamento na Web Geográfica: Serviços de processamento*

G. Arquiteturas e aplicações Para a Web Geográfica: Arquiteturas; Globos virtuais; Sistemas de Mapas para a Web; Sistemas Baseados na Localização

H. Interfaces e interação na Web Geográfica: Interfaces Cartográficos; Geovisualização; Tecnologias de interação para SIGs Móveis: Realidade Virtual e Aumentada

4.4.5. Syllabus:

- A. Introduction to Geographic Information (GI): Nature and functionality of GI
- B. Representing Geographic Information: Field-based models; Entity-based models; Data Structures for Geographic Information
- C. Georeferencing: Coordinate Systems and Projection Systems (examples)
- D. The GeoWeb: Concept, current capacity; Vision for the future
- E. The GeoWeb: Distributing the data; Standards for geographic data; Geographic Metadata; Geographic databases
- F. The GeoWeb: Distributing the Software; Processing Services for the GeoWeb

- G. Architectures and applications for the GeoWeb; Multi-tier architectures; Virtual Globes; Web Mapping Systems; Location-based Systems
- H. User interfaces and interaction in the GeoWeb; Cartographic Interfaces; Geo-visualization; Interaction Technology for Mobile GIS: Virtual and Augmented Reality

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

- O objetivo 1 é endereçado nos capítulos D, E, F, G e H do programa.
- O objetivo 2 é endereçado no capítulo A, B, C e E do programa.
- O objetivo 3 é endereçado no capítulo E, F, G, e H do programa.
- Os capítulos C, E, F, G e H do programa têm relevância para o objetivo 4.
- Todos os capítulos podem ter relevância para o objetivo 5.

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

- Objective 1 is addressed in chapters D, E, F, G, and H of the syllabus.
- Objective 2 is addressed in chapter A, B, C, and E of the syllabus.
- Objective 3 is addressed in chapter E, F, G, and H of the syllabus.
- Chapters C, E, F, G, and H are of relevance in objective 4.
- All chapters are of relevance in objective 5.

4.4.7. Metodologias de ensino (avaliação incluída):

Os conteúdos teóricos são endereçados em aulas teóricas, sendo instanciados com exercícios e exemplos nas aulas práticas que vão construindo um corpo de técnicas para o desenvolvimento do trabalho prático. Esta componente é avaliada através de 2 testes, valendo cada um 30% da nota (NT).

Os alunos têm que desenvolver um trabalho prático, em grupos, em que concebem um sistema que contempla um conjunto de requisitos de base na Web Geográfica. Este trabalho deve ser objeto de escrita de relatório e apresentação oral. A nota do trabalho vale 40% da nota (NP).

Para obter aprovação à disciplina, os alunos devem conseguir uma nota mínima de 9,5 (em 20) em NT e NP. Em época de recurso, a nota dos dois testes é substituída pela nota do exame, que vale 60% (NT) da nota final.

4.4.7. Teaching methodologies (including students' assessment):

Theoretical issues are introduced in lectures, and instantiated in exercises and examples in labs, as the students build a body of techniques to support the development of the practical assignment. This component is evaluated through 2 tests, each one worth 30% of the final grade (NT).

Students must develop a group assignment in which they conceive a system contemplating a set of fundamental requirements for the GeoWeb, and which will provide access to the exam. These students must write a report describing their system and present their work. The assignment grade is worth 40% of the final grade (NP).

In the appeal season, the tests grade is replaced by the exam grade, which is worth 60% (NT) of the final grade. A student is approved if both NP and NT are greater or equal to 9.5 points out of 20.

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

As aulas teóricas seguem todo o programa, tendo em conta os objetivos 1 a 4.

Nas aulas práticas são necessários os conhecimentos obtidos nas aulas teóricas (objs. 1 a 4) mas os exercícios e exemplos considerados estão diretamente relacionados com o objetivo 5, construindo os alunos um conjunto de capacidades de manipulação de ferramentas que lhe permitem depois atingir o objetivo 6.

O desenvolvimento do trabalho prático implica o estudo de um domínio de aplicação, assim como a avaliação das necessidades dos utilizadores nesse contexto (obj. 10).

Durante o desenvolvimento do trabalho prático, os alunos usam os conhecimentos adquiridos nas aulas teóricas e necessitam de ser conhecedores das ferramentas usadas nas aulas práticas. No entanto, precisam também de ser capazes de gerir o tempo e organizarem-se (objs.7). O trabalho prático também implica a escrita de um relatório e uma apresentação oral (objs.8).

Finalmente, nos testes ou exame, são endereçados os objetivos 1 a 4 e 7.

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

Lectures go over the content of the syllabus, addressing objectives 1 through 4.

Labs take into account what has been introduced in the lectures, thus addressing the objectives 1 through 4. Moreover, during the labs, exercises and examples address objective 5 and build up to endow students with tools to achieve objective 6.

The development of the practical assignment involves analyzing an application domain and evaluating users' needs in that context (objective 10).

While developing the practical assignment, the students must be aware of the content provided in the lectures and be proficient in the use of the tools used in the labs, but they must also be able to manage time and be organized (objective 7). The practical assignment also involves writing a report and orally presenting their work, and this is

*related with objective 8.
In the tests or exam, objectives 1-4 and 7 are addressed.*

4.4.9. Bibliografia de consulta/existência obrigatória:

- Longley, P. A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W., *Geographic Information Systems and Science, 4rd Edition, John Wiley and Sons, 2014.*
- Worboys, M. and Duckham, M. *GIS: A Computing Perspective, 2nd Edition, CRC Press, 2004.*
- Wade Bishop and Tony H. Grubestic, *Geographic Information: Organisation, Access, and Use, Springer, 2016.*

Mapa IV - Dissertação em Engenharia Informática**4.4.1.1. Designação da unidade curricular:**

Dissertação em Engenharia Informática

4.4.1.1. Title of curricular unit:

Dissertation in Computer Science and 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:

840

4.4.1.5. Horas de contacto:

OT:40

4.4.1.6. ECTS:

30

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Carlos Augusto Isaac Piló Viegas Damásio - OT:40

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

Todos os docentes do curso de Mestrado em Engenharia Informática

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

O trabalho de dissertação de Mestrado em Engenharia Informática enquadra-se nos termos definidos na alínea b) do número 1 do Artigo 20º do Decreto Lei 65/2018, e consiste num trabalho individual de investigação e/ou desenvolvimento que explora os conhecimentos adquiridos ao longo do curso, original e especialmente concebido para esse fim. O trabalho de dissertação baseia-se na elaboração de uma tese de natureza científica.

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

The dissertation work for the Master in Computer Science and Engineering conforms with the requirements of item b) of number 1, Art 20º of DL 65/2018, and consists of an individual research and / or development work which explores the knowledge acquired in the whole program, original, and specifically identified for that purpose. The dissertation work is based on the elaboration of a research-oriented thesis of scientific nature.

4.4.5. Conteúdos programáticos:

Desenvolvimento de trabalho conducente a elaboração de dissertação de mestrado em Engenharia Informática.

4.4.5. Syllabus:

Development of the work leading to the elaboration of a master dissertation in Computer Science and Engineering.

4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:
Os conteúdos são naturalmente coerentes com os objetivos, dada a sua natureza especial da unidade curricular.

4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:
Given the special nature of the curricular unit, the syllabus are naturally coherent with the unit's objectives

4.4.7. Metodologias de ensino (avaliação incluída):

O trabalho conducente à dissertação de mestrado desenrola-se em duas fases. Na primeira fase, desenvolvida no contexto da UC Preparação de Dissertação / Projeto de Engenharia Informática o estudante desenvolve o trabalho introdutório do tema da sua dissertação e apresenta um relatório incluindo um plano do trabalho a realizar para concluir a Dissertação de Mestrado.

Nesta UC, com a duração de 1 semestre, 30 ECTS, o estudante desenvolve o trabalho de dissertação, de acordo com o plano elaborado na UC de preparação. O trabalho desenvolvido será formalmente apresentado na Dissertação de Mestrado, defendida em provas públicas perante o júri, que a avalia e lhe atribui uma nota final, de acordo com o regulamento geral da FCT e da UNL.

4.4.7. Teaching methodologies (including students' assessment):

The work leading to the master dissertation develops in two consecutive phases. In the first phase, in the context of CU Preparation of Dissertation / Project in Computer Science and Engineering, the student develops preliminary work for the dissertation theme and presents a written report, including a plan of the work needed to conclude the master dissertation work.

In this CU, with the duration of one term, 30 ECTS, the student develops the dissertation work, according to the plan devised in the preparation CU. The work developed is formally reported in the Master Dissertation, which is defended by the candidate in public exam before a jury who assigns the final grading, according to the general regulations of FCT and UNL.

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

Tendo em conta a natureza especial desta unidade curricular, decorre que existe coerência adequada entre as metodologias de ensino e os objetivos de aprendizagem da mesma. Acresce que a metodologia de ensino adotada, estruturada em duas UC, com um ponto de aferição no fim da UC de Preparação de Dissertação /Projeto em Engenharia Informática, e que resulta da experiência de vários anos do Departamento de Informática na orientação de dissertações de mestrado, tem-se revelado um fator promotor do aumento de qualidade do processo de desenvolvimento das dissertações, na opinião de todos os envolvidos (estudantes, docentes, membros de júri, e parceiros exteriores).

A segunda fase do trabalho, desenvolvida no contexto desta UC, correspondente à elaboração da dissertação e estimada em 30 ECTS será tipicamente desenvolvida pelo estudante em dedicação total. O trabalho de dissertação poderá desenvolver-se quer em contexto académico, quer em contexto de colaboração académico-empresarial, em qualquer dos casos, sob coordenação científica geral da comissão científica de curso e orientação formal de um membro do corpo docente do curso. Tipicamente, serão avaliados os seguintes parâmetros:

- 1. Qualidade da análise do estado da arte e da sua adequação aos objetivos do trabalho;*
- 2. Qualidade do trabalho realizado;*
- 3. Extensão do trabalho realizado;*
- 4. Qualidade da análise crítica da contribuição e dos resultados obtidos;*
- 5. Qualidade da apresentação oral e da argumentação;*
- 6. Qualidade do Relatório;*
- 7. Apreciação Geral.*

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

Having in mind the special nature of this curricular unit, it clearly results that there is adequate coherence between the teaching methodologies and the learning outcomes. Additionally, we note that the teaching methodology, structured in two phases, with an interim assessment point in the end of the CU Preparation of Dissertation / Project in Computer Science Engineering, which results from the several years of experience of the Department on supervising master dissertations, has revealed itself as a factor that promotes a quality increase of the dissertation supervision process, in the opinion of all the stakeholders (students, faculty members, jury members, and external collaborators).

The second part of the work, developed in the context of this CU, corresponds to the elaboration of the planned dissertation work with an estimated work of 30 ECTS and it is typically developed in full time dedication by the student. The dissertation work can be developed either in academic context or in some collaboration between the department and an external institution (such as a firm), under the scientific coordination of the program's scientific committee and formal supervision of a faculty member. Typically, the following parameters are taken into account:

- 1. Quality of the state-of-the-art review and its suitability to the work objectives;*
- 2. Quality of the performed work;*
- 3. Extension of the performed work;*
- 4. Quality of the critical analysis of the contributions and results obtained;*
- 5. Quality of the oral presentation and argumentation;*
- 6. Quality of the written document;*
- 7. Global appreciation.*

4.4.9. Bibliografia de consulta/existência obrigatória:

Depende de cada trabalho de dissertação / Depends on each specific dissertation work.

Mapa IV - Projeto em Engenharia Informática**4.4.1.1. Designação da unidade curricular:**

Projeto em Engenharia Informática

4.4.1.1. Title of curricular unit:

Project in Computer Science and 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:

840

4.4.1.5. Horas de contacto:

OT:40

4.4.1.6. ECTS:

30

4.4.1.7. Observações:

Optativa

4.4.1.7. Observations:

Optional

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

Carlos Augusto Isaac Piló Viegas Damásio - OT:40

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

Todos os docentes do curso de Mestrado em Engenharia Informática

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

O trabalho de dissertação de Mestrado (Dissertação Científica) enquadra-se nos termos definidos na alínea b) do número 1 do Artigo 20º do Decreto Lei 65/2018, e consiste num trabalho individual de investigação e/ou desenvolvimento que explora os conhecimentos adquiridos ao longo do curso, original e especialmente concebido para esse fim. O trabalho de Projeto em Engenharia Informática baseia-se no desenvolvimento de um projeto avançado de engenharia de conceção.

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

The dissertation work for the Master in Computer Science and Engineering conforms with the requirements of item b) of number 1, Art 20º of DL 65/2018, and consists of an individual research and / or development work which explores the knowledge acquired in the whole program, original, and specifically identified for that purpose. The dissertation work is based on the development of an advanced conception-oriented engineering project.

4.4.5. Conteúdos programáticos:

Desenvolvimento de trabalho conducente a elaboração do Projeto em Engenharia Informática.

4.4.5. Syllabus:

Development of the work of the Project in Computer Science and Engineering.

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

Os conteúdos são naturalmente coerentes com os objetivos, dada a sua natureza especial da unidade curricular.

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

Given the special nature of the curricular unit, the syllabus are naturally coherent with the unit's objectives

4.4.7. Metodologias de ensino (avaliação incluída):

O trabalho conducente ao desenvolvimento do projeto em Engenharia Informática desenrola-se em duas fases. Na primeira fase, desenvolvida no contexto da UC Preparação de Dissertação / Projeto de Engenharia Informática o estudante desenvolve o trabalho introdutório do tema da sua dissertação e apresenta um relatório incluindo um plano do trabalho a realizar para concluir o projecto de mestrado.

Nesta UC, com a duração de 1 semestre, 30 ECTS, o estudante desenvolve o trabalho de Projeto em Engenharia Informática, de acordo com o plano elaborado na EC de preparação. O trabalho desenvolvido será formalmente apresentado num Dissertação de Mestrado, defendida em provas públicas perante o júri, que a avalia e lhe atribui uma nota final, de acordo com o regulamento geral da FCT e da UNL.

4.4.7. Teaching methodologies (including students' assessment):

The work of the project in Computer Science and Engineering has two consecutive phases. In the first phase, in the context of CU Preparation of Dissertation / Project in Computer Science and Engineering, the student develops preliminary work for the project theme and presents a written report, including a plan of the work needed to conclude the master project work.

In this CU, with the duration of one term, 30 ECTS, the student develops the project work, according to the plan devised in the preparation CU. The work developed is formally reported in the Master Dissertation, which is defended by the candidate in public exam before a jury who assigns the final grading, according to the general regulations of FCT and UNL.

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

Tendo em conta a natureza especial desta unidade curricular, decorre que existe coerência adequada entre as metodologias de ensino e os objetivos de aprendizagem da mesma. Acresce que a metodologia de ensino adotada, estruturada em duas UC, com um ponto de aferição no fim da UC de Preparação de Dissertação /Projeto em Engenharia Informática, e que resulta da experiência de vários anos do Departamento de Informática na orientação de dissertações de mestrado, tem-se revelado um fator promotor do aumento de qualidade do processo de desenvolvimento das dissertações, na opinião de todos os envolvidos (estudantes, docentes, membros de júri, e parceiros exteriores).

A segunda fase do trabalho, desenvolvida no contexto desta UC, correspondente à elaboração do projeto de Engenharia em Engenharia Informática e estimada em 30 ECTS será tipicamente desenvolvida pelo estudante em dedicação total. O trabalho de projeto será tipicamente desenvolvido em ambiente empresarial, sob coordenação científica geral da comissão científica de curso e orientação formal de um membro do corpo docente do curso e de um elemento da empresa. Tipicamente, serão avaliados os seguintes parâmetros:

- 1. Qualidade da análise do estado da arte e da sua adequação aos objetivos do trabalho;*
- 2. Qualidade do trabalho realizado;*
- 3. Extensão do trabalho realizado;*
- 4. Qualidade da análise crítica da contribuição e dos resultados obtidos;*
- 5. Qualidade da apresentação oral e da argumentação;*
- 6. Qualidade do Relatório;*
- 7. Apreciação Geral.*

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

Having in mind the special nature of this curricular unit, it clearly results that there is adequate coherence between the teaching methodologies and the learning outcomes. Additionally, we note that the teaching methodology, structured in two phases, with an interim assessment point in the end of the CU Preparation of Dissertation / Project in Computer Science Engineering, which results from the several years of experience of the Department on supervising master dissertations, has revealed itself as a factor that promotes a quality increase of the supervision process, in the opinion of all the stakeholders (students, faculty members, jury members, and external collaborators).

The second part of the work, developed in the context of this CU, corresponds to the elaboration of the planned project work with an estimated work of 30 ECTS and it is typically developed in full time dedication by the student. The project work is typically developed in the context of an external organization (such as a firm), under the scientific coordination of the program's scientific committee and formal supervision of a faculty member and an element of the external organization. Typically, the following parameters are taken into account:

- 1. Quality of the state-of-the-art review and its suitability to the work objectives;*
- 2. Quality of the performed work;*
- 3. Extension of the performed work;*
- 4. Quality of the critical analysis of the contributions and results obtained;*
- 5. Quality of the oral presentation and argumentation;*
- 6. Quality of the written document;*
- 7. Global appreciation.*

4.4.9. Bibliografia de consulta/existência obrigatória:

Depende de cada trabalho de dissertação / Depends on each specific dissertation work.

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:**

O curso permite flexibilidade no percurso dos estudantes, com um bloco base - que obriga a realização de UC de várias áreas, e um bloco de especialização em que se pode optar entre concentrar a formação numa sub-área ou aprofundar temas à escolha, numa perspetiva de abrangência. Esta organização permite adquirir um conhecimento dos fundamentos e métodos da Engenharia Informática de Conceção, com foco numa dada área. A articulação entre a vertente teórica e prática nas UC garante que os estudantes adquirem um conjunto de aptidões e competências para colocarem estes conhecimentos em prática.

Os desafios colocados pelos trabalhos das UC e pela Dissertação exigem que os alunos lidem com problemas de diferentes graus de complexidade e que os resolvam com soluções criativas, fomentando a sua autonomia; as UC fomentam uma apresentação crítica dos trabalhos efetuados nos relatórios dos mesmos.

Os aspetos ético-profissionais da informática e do empreendedorismo são abordados em UC obrigatórias.

4.5.1. Evidence of the teaching and learning methodologies coherence with the intended learning outcomes of the study programme:

The programme allows for flexibility in the students' academic record, with a core block that requires them to take courses from different areas, and a specialization block to concentrate the education in a subarea or deepen the knowledge in topics of choice, covering different subareas. This organization allows the acquisition of the foundations and methods necessary for conception-oriented engineering in CSE, focusing on a given area. The balance between the theoretical and practical aspects ensures the acquisition of the set of skills and competences to put this knowledge into practice. The challenges posed by the projects in the UC and the Dissertation require students to deal with problems of varying degrees of complexity and to solve them with creative solutions, fostering their autonomy; the courses foster the critical presentation of the works done in the presentations and reports produced. The ethical-professional aspects and entrepreneurship are tackled in mandatory courses.

4.5.2. Forma de verificação de que a carga média de trabalho que será necessária aos estudantes corresponde ao estimado em ECTS:

Para o 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 UC tem sido continuamente aferido e quando necessário reajustado pelos docentes e comissão científica do curso, de forma informada por inquéritos periódicos na FCT NOVA.

Na presente proposta foi de novo analisada a creditação das UC, tendo sido preocupação equilibrar o esforço do estudante entre os vários semestres do curso e reavaliar o esforço das várias componentes de cada UC. 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. The student load in the various components of activity of each course has been continuously monitored and adjusted whenever considered necessary by the teaching staff and programme scientific committee, informed by periodic student surveys at FCT NOVA.

When preparing this proposal, the ECTS for all courses was again reconsidered, with a special concern to better balance the students' work load along the several terms of the programme. According to the general recommendations of FCT NOVA, all courses 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 ficha de cada UC apresenta os objetivos da aprendizagem, definidos em conhecimentos, aptidões e competências a adquirir pelos estudantes, assim como as metodologias de ensino, incluindo a avaliação da UC. Adicionalmente, cada ficha inclui uma demonstração de como as metodologias de ensino, incluindo as componentes teóricas e práticas, permitem alcançar os diversos objetivos de aprendizagem. A avaliação das várias componentes permite avaliar o nível de aprendizagem dos alunos relativamente aos diversos objetivos de aprendizagem, nomeadamente em termos de conhecimentos, aptidões e competências.

A comissão científica monitora o funcionamento das UC e coordena os momentos da avaliação de cada UC, em colaboração com a comissão pedagógica, que inclui alunos do curso. Adicionalmente, usa a informação dos inquéritos aos alunos (promovidos pela DPGQ), os quais incluem questões relativas à avaliação, para identificar situações que necessitem de ajustes.

4.5.3. Means of ensuring that the students assessment methodologies are adequate to the intended learning outcomes:

The course description of each UC presents the learning objectives, defined in knowledge, skills and competences to be acquired by students, as well as the teaching methodologies, including students' assessment rules. Additionally, each course description includes a demonstration of how teaching methodologies, including theoretical and practical components, enable to achieve the course's learning objectives. The assessment of the different components of each course allows one to assess the level of learning in relation to the learning objectives, in particular in terms of knowledge, skills and competences.

The scientific committee monitors the curricular units and coordinates the evaluation moments of each curricular unit, in collaboration with the pedagogical committee, which includes students of the programme. In addition, it uses information from student surveys (promoted by DPGQ), which include questions regarding students' assessment, to identify situations requiring adjustment.

4.5.4. Metodologias de ensino previstas com vista a facilitar a participação dos estudantes em atividades científicas (quando aplicável):

As UC Preparação de Dissertação e Dissertação permitem aos alunos participar em atividades científicas, efetuando um trabalho de investigação tipicamente no contexto dos projetos em curso no centro de investigação NOVA LINCS. A separação da dissertação em duas UC tem-se revelado muito positiva, permitindo aos alunos, na fase de Preparação, familiarizarem-se com o tópico no qual trabalharão, através dum levantamento do estado da arte.

As UC de especialização abordam tópicos avançados, permitindo aos alunos o contacto com resultados científicos recentes. A componente prática das UC permite aos alunos contactar com as bases do processo científico nas diferentes áreas, a qual pode incluir, por exemplo, a criação dum protótipo e sua avaliação ou a definição das propriedades dum algoritmo e sua prova.

O DI tem um programa de bolsas para alunos de mestrado com componentes letiva e de investigação, que visa fomentar a participação dos melhores alunos em atividades de investigação.

4.5.4. Teaching methodologies that promote the participation of students in scientific activities (as applicable):

The Preparation and Dissertation/Project UC allow students to participate in scientific research activities, carrying out research work typically in on-going projects at the NOVA LINCS research centre. The split of the dissertation in two courses has been very positive, allowing students, in the Preparation course, to familiarize themselves with the topic on which they will work, through a state of the art survey. Specialization courses address advanced topics, introducing students to recent scientific results. The practical component of courses allows students to familiarize themselves with the practice of the scientific process in different areas, which may include, for example, the creation of a prototype and its evaluation or the definition of the properties of an algorithm and its proof. The department has a scholarship program for MSc students that includes a teaching and a research component, which aims to encourage the participation of the best students in research activities.

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 ciclo de estudos de Mestrado (2.º ciclo), com 2 anos (4 semestres), foi atribuído ao ciclo de estudos um total de 120 ECTS para a obtenção do grau de Mestre. O mestrado é concluído com uma Dissertação de Mestrado ou Projeto de Engenharia com um total de 30 ECTS. Em ambos os casos, este trabalho é precedido por uma Preparação de Dissertação/Projeto com um total de 12 ECTS.

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 19º of Decreto-Lei no 74/2006, being a Master programme (2nd cycle), with 2 years (4 semesters), we assign to the study cycle a total of 120 ECTS for obtaining the Master degree. The master programme is concluded with a Master's Dissertation or an Engineering Project with a total of 30 ECTS. In both cases, this work is preceded by a Dissertation/Project Preparation with a total of 12 ECTS.

4.6.2. Forma como os docentes foram consultados sobre a metodologia de cálculo do número de créditos ECTS das unidades curriculares:

O esforço do estudante nas várias componentes de atividade de cada UC do curso de Mestrado em Engenharia Informática que se propõe, e que na maioria dos casos resultam de adaptação de UC que transitaram dos 4.º e 5.º anos do atual curso de Mestrado Integrado em Engenharia Informática, tem sido continuamente aferido e reajustado pelos docentes das mesmas sob a coordenação da Comissão Científica, de forma informada por inquéritos na FCT NOVA. 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 cálculo dos ECTS. As fichas das UC incluem uma distribuição do número de horas a serem utilizadas em cada componente da atividade da UC.

4.6.2. Process used to consult the teaching staff about the methodology for calculating the number of ECTS credits of the curricular units:

The student's effort in the various activity components of each curricular unit of the Master programme now proposed, which in most cases result from the adaptation of courses of the 4th and 5th year of the current Integrated Master programme in Computer Science and Engineering, has been continuously monitored and adjusted by the academic staff under the coordination of the Scientific Committee, informed by the periodic student surveys carried out at FCT NOVA. These surveys consult students about their workload in several academic activities and are considered in the process of adjusting ECTS credits.

During the process of elaboration of this proposal and of the courses' descriptions included in this proposal, the academic staff was actively involved in the calculation of ECTS of each course. The course description includes a distribution of the number of hours to be used of in each component of the course activity.

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, 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, 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 e Carlos Augusto Isaac Piló Viegas Damásio

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
José Augusto Legatheaux Martins	Professor Catedrático ou equivalente	Doutor	Informática - Sistemas Distribuídos	100	Ficha submetida
Luís Manuel Marques da Costa Caires	Professor Catedrático ou equivalente	Doutor	Informática	100	Ficha submetida
Nuno Manuel Robalo Correia	Professor Catedrático ou equivalente	Doutor	Eng. Electr. e Comp.	100	Ficha submetida
Pedro Manuel Corrêa Calvente Barahona	Professor Catedrático ou equivalente	Doutor	Informática (Computer Science)	100	Ficha submetida
José Júlio Alves Alferes	Professor Catedrático ou equivalente	Doutor	Informática	100	Ficha submetida
João Alexandre Carvalho Pinheiro Leite	Professor Associado ou equivalente	Doutor	Informática	100	Ficha submetida
Ana Maria Diniz Moreira	Professor Associado convidado ou equivalente	Doutor	Engenharia de Software/Engenharia Informática	20	Ficha submetida
Carmen Pires Morgado	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	Ficha submetida
Nuno Manuel Ribeiro Preguiça	Professor Associado ou equivalente	Doutor	Informática	100	Ficha submetida
João Miguel da Costa Magalhães	Professor Associado ou equivalente	Doutor	Doutoramento em Computer Science	100	Ficha submetida
Sofia Carmen Faria Maia Cavaco	Professor Auxiliar ou equivalente	Doutor	Informática	60	Ficha submetida
Vasco Miguel Moreira do Amaral	Professor Auxiliar ou equivalente	Doutor	Ciência da Computação	100	Ficha submetida
Carlos Augusto Isaac Piló Viegas Damásio	Professor Associado ou equivalente	Doutor	Informática	100	Ficha submetida
Carla Maria Gonçalves Ferreira	Professor Associado ou equivalente	Doutor	PhD in Computer Science	100	Ficha submetida
António Carlos Bárbara Grilo	Professor Associado ou equivalente	Doutor	Gestão Industrial - Comércio Electrónico	100	Ficha submetida
António Maria Lobo César Alarcão Ravara	Professor Associado ou equivalente	Doutor	Matemática	100	Ficha submetida
Bernardo Parente Coutinho Fernandes Toninho	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Fernando Pedro Reino da Silva Birra	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Henrique João Lopes Domingos	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Hervé Miguel Cordeiro Paulino	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
João Baptista da Silva Araújo Júnior	Professor Auxiliar ou equivalente	Doutor	Software Engineering	100	Ficha submetida
João Carlos Gomes Moura Pires	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
João Manuel dos Santos	Professor Auxiliar ou	Doutor	Informática	100	Ficha

Lourenço	equivalente					submetida
João Ricardo Viegas da Costa Seco	Professor Auxiliar ou equivalente	Doutor	Informática	100		Ficha submetida
Joaquim Francisco Ferreira da Silva	Professor Auxiliar ou equivalente	Doutor	Informática	100		Ficha submetida
Jorg Matthias Knorr	Professor Auxiliar ou equivalente	Doutor	Doutoramento em Informática	100		Ficha submetida
Jorge Carlos Ferreira Rodrigues da Cruz	Professor Auxiliar ou equivalente	Doutor	Informática	100		Ficha submetida
Margarida Paula Neves Mamede	Professor Auxiliar ou equivalente	Doutor	Informática	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
Maria Armada Simenta Rodrigues Grueau	Professor Auxiliar ou equivalente	Doutor	Information Science	100		Ficha submetida
Ludwig Krippahl	Professor Auxiliar ou equivalente	Doutor	Bioquímica estrutural	100		Ficha submetida
Miguel Jorge Tavares Pessoa Monteiro	Professor Auxiliar ou equivalente	Doutor	Informática	100		Ficha submetida
Paulo Orlando Reis Afonso Lopes	Professor Auxiliar ou equivalente	Doutor	Informática	100		Ficha submetida
Teresa Isabel Lopes Romão	Professor Auxiliar ou equivalente	Doutor	Engenharia do Ambiente	100		Ficha submetida
Susana Maria dos Santos Nascimento Martins de Almeida	Professor Auxiliar ou equivalente	Doutor	Informática	100		Ficha submetida
Sérgio Marco Duarte	Professor Auxiliar ou equivalente	Doutor	Informática	100		Ficha submetida
João Carlos Antunes Leitão	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática e de Computadores	100		Ficha submetida
Ruy Araújo da Costa	Professor Associado ou equivalente	Doutor	Engenharia de Sistemas	100		Ficha submetida
				3780		

<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.

39

5.4.1.2. Número total de ETI.

37.8

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:	37	97.883597883598

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):	37.8	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	35.8	94.708994708995 37.8
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 37.8

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	37	97.883597883598 37.8
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 37.8

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 lecionação 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 lecionação 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 Gabinetes e salas para pessoal administrativo e técnico (área 170 m2/ocupação de 9 pessoas)

6 Salas servidores e de infra-estrutura informática (área total 100m2)

8 Laboratórios de Investigação (área 300m2/ocupação máxima 73 colaboradores)

10 Laboratórios de Ensino Informática com datashow (área 700m2/300 lugares - 30 por sala), incluindo laboratórios específicos de redes e projectos de alunos.
 6 Salas alunos de utilização livre com computador ou mesa de trabalho (área 170m2/88 lugares)
 34 Infraestrutura de controle de acessos suportada em cartões magnéticos.
 10 Câmaras de video-vigilância
 Vários espaços comuns; Biblioteca central da FCT NOVA, Cafetarias, 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 e videoconferencing (area 80 m2/capacity 20 people)
 1 Seminar room with datashow e 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 e 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 Equipamento data show (fixo)
 3 Equipamento data show (portátil)
 190 Computadores (para alunos)
 70 Computadores (docentes)
 10 Computadores (serviços técnicos e administrativos)
 19 Computadores (serviços centrais)
 7 Computadores agregados em cluster (alunos)
 17 Computadores agregados em cluster (investigação)
 17 Infraestruturas de rede sem fios (APs)
 34 Infraestruturas de rede com fios (switches)
 2 Firewalls
 4 Impressoras para docentes, serviços e pós-graduação
 2 Fotocopiadora
 2 Equipamento de video-conferência; 1 Sistema de som
 2 Televisor plasma
 15 Computador portáteis tipo tablet
 60 Monitores TFT 22" (para ligação de portáteis)

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 desktop (for students)
 70 Computer desktop/laptop (teaching staff)
 10 Computer desktop(technical and administrative staff)
 19 Computer workstations (central services)
 7 Servers in cluster configuration (students)
 17 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 - Laboratory for Computer Science and Informatics	Excelente / Excellent	Faculdade de Ciências e Tecnologia da Universidade NOVA de Lisboa	36	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.

<https://a3es.pt/si/iportal.php/cv/scientific-publication/formId/42cb9283-2f0d-8096-7654-5e71164a97b7>

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:

<https://a3es.pt/si/iportal.php/cv/high-level-activities/formId/42cb9283-2f0d-8096-7654-5e71164a97b7>

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:

De acordo com os Dados e Estatísticas de Cursos Superiores disponíveis em dezembro de 2019 no Portal Infocursos da DGES, para estudantes diplomados entre 2013/14 e 2016/17, encontravam-se registados como desempregados no Instituto de Emprego e Formação Profissional 2% dos diplomados no curso de Mestrado Integrado em Engenharia Informática, correspondente a uma taxa de empregabilidade de 98%. De acordo com os mesmos dados, a média nacional dos cursos da área é de 1,5% de diplomados, correspondente a uma taxa de empregabilidade de 98,5%.

De acordo com os dados recolhidos pela Universidade NOVA de Lisboa e coligidos no relatório do OBIPNOVA (Inquéritos efetuados em 2016 aos Diplomados de 2014 (1 ano após)), a percentagem de diplomados com o curso de Mestrado Integrado em Engenharia Informática da FCT NOVA com emprego ao fim de um ano é de 100%.

Desde 1999, o Departamento de Informática da FCT NOVA graduou mais de 1700 licenciados e mestres em Engenharia Informática.

9.1. Evaluation of the employability of graduates by similar study programmes, based on official data:

According to the “Data and Statistics on Higher Education” available in December of 2019 at DGES Infocursos Portal, for students graduated between 2013/14 and 2016/17, the unemployment rate for students of the FCT NOVA Integrated Master in Computer Science and Engineering was 2% according to data from “Instituto de Emprego e Formação Profissional”, corresponding to an employment rate of 98%. According to the same data, the national average of degrees in the area was 1.5% of graduates, corresponding to an employability rate of 98.5%.

According to data collected by NOVA and collected in the OBIPNOVA report (surveys conducted in 2016 to students graduated in 2014), the percentage of graduates with the FCT NOVA Integrated Master in Computer Science and Engineering with employment, one year after graduation, was 100%.

Since 1999, FCT NOVA’s Department of Informatics has delivered more than 1700 “licenciados” and master level graduates Computer Science and Engineering.

9.2. Avaliação da capacidade de atrair estudantes baseada nos dados de acesso (DGES):

A 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. De acordo com a DGES, o número de candidatos que nos últimos anos manifestou interesse no Mestrado Integrado em Engenharia Informática (MIEI) da FCT NOVA tem sido sempre muito superior ao número de vagas, com a nota do último candidato admitido no contingente geral a ser superior a 14,9 nos últimos quatro anos (desde 2016/17). A percentagem de candidatos admitidos com notas superiores a 16 tem aumentado, tendo sido superior a 43% nos últimos três anos. Os candidatos admitidos obtêm em grande maioria a sua colocação em primeira opção.

Nos últimos quatro anos transitaram ou ingressaram no 2.º ciclo do MIEI 150 alunos por ano, em média. Conjugando estes dados com os altos níveis de empregabilidade dos cursos de Mestrado na área da Informática, acredita-se que o curso proposto apresenta um alto potencial de atração.

9.2. Evaluation of the capability to attract students based on access data (DGES):

FCT NOVA has demonstrated for decades a sustained ability to attract candidates for its educational offer in Computer Science and Engineering in both quality and quantity. According to DGES, the number of candidates that demonstrated interest in the FCT NOVA Integrated Master in Computer Science and Engineering (MIEI) has been much higher than the number of available positions, with the application grade of the last student enrolled being higher than 14.9 in the last 4 years. The percentage of candidates enrolled with grade higher than 16 has been larger than 43% in the last 3 years. Most of the candidates enrolled have selected the NOVA MIEI as their first option.

Over the past 4 years, more than 150 students per year have transitioned or enrolled in the 2nd cycle of MIEI.

Considering this data and the high level of employment of the Masters degrees in the area of Computer Science and Engineering, we believe that the proposed programme presents an 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 2.º ciclo (e do 1.º ciclo) não estão estabelecidas parcerias formais com instituições da região. No entanto, o Departamento de Informática mantém colaborações estreitas com escolas congéneres na região como o Instituto Superior Técnico 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, Beira Interior e Madeira) 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:

Concerning the education at the 2nd cycle (and also at the 1st cycle), we have no established partnerships with other local institutions. However, our Department is developing close collaboration with nearby engineering and science higher education schools such as the Instituto Superior Técnico and Faculdade de Ciências of Universidade de Lisboa, for example, in the evaluation of master dissertations, research project collaborations, and actively sustains and cherishes open communication and interchange channels.

The Department also maintains close collaboration with other institutions (Universidade de Évora, Universidade da Beira Interior e Universidade da Madeira) through the participation of academic and research staff from those universities in the NOVA LINCS research centre.

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:

O curso proposto compara-se naturalmente com os programas de escolas europeias de referência que fornecem um curso de mestrado que visa a formação de engenheiros informáticos de conceção. Por exemplo, a Universidade Federal de Lausane (EPFL) e a Technical University of Munich (TUM) fornecem cursos de mestrado de 2 anos (120 ECTS), com uma estrutura semelhante ao do curso proposto, em que os alunos devem selecionar UC de um conjunto de UC base e de especialização. Nestes cursos, a dissertação tem apenas 30 ECTS. Outras universidades, como a Universidade Federal de Zurique (ETHZ) e o Imperial College oferecem cursos de mestrado mais curtos, com apenas 90 ECTS.

Em Portugal, é comum o modelo de mestrado com 120 ECTS, o qual é oferecido, por exemplo, pelo IST da

Universidade de Lisboa e pela FCT da Universidade de Coimbra. A FCT NOVA também ofereceu um curso de mestrado de 120 ECTS antes de oferecer o Mestrado Integrado em Engenharia Informática.

10.1. Examples of study programmes with similar duration and structure offered by reference institutions in the European Higher Education Area:

The proposed programme naturally compares with programmes of reference European schools targeting the education of conception-oriented engineer in Computer Science and Engineering. For example, the Federal University of Lausanne (EPFL) and the Technical University of Munich (TUM) offer 2-year Masters programmes (120 ECTS), with a similar structure to the proposed programme, where students must select courses from a set of base and specialization courses. In these degrees, the dissertation has only 30 ECTS. Other universities, such as the Federal University of Zurich (ETHZ) and Imperial College, offer shorter Masters programmes with only 90 ECTS.

In Portugal, Master programmes with 120 ECTS are the most common model, which is offered, for example, by IST of U. Lisbon and by FCT of University of Coimbra. FCT NOVA also offered a 120 ECTS Master programme before offering the Integrated Master in Computer Science and Engineering.

10.2. Comparação com objetivos de aprendizagem de ciclos de estudos análogos existentes em instituições de referência do Espaço Europeu de Ensino Superior:

Os objetivos de aprendizagem dos ciclos de estudo análogos são genericamente semelhantes, focando-se em fornecer os conhecimentos e aptidões necessárias a um Engenheiro Informático de conceção. Para tal, as UC do bloco base abordam um subconjunto de tópicos core e opcionais dos currícula ACM/IEEE, permitindo obter uma formação alargada por terem necessariamente de realizar UC em diversas áreas. O bloco de especialização tende a ter uma natureza mais variada. Existem cursos em que se procura o aprofundamento de uma dada vertente. Outros, como nesta proposta, permitem ao estudante optar entre concentrar a formação numa subárea ou aprofundar assuntos à sua escolha, numa perspetiva de abrangência.

Alguns ciclos de estudo (e.g EPFL) incluem um estágio obrigatório entre 8 semanas e 6 meses, podendo ser combinado com a dissertação de mestrado. No ciclo de estudos proposto existe a possibilidade de concluir o curso com um Projeto de Engenharia com objetivos semelhantes.

10.2. Comparison with the intended learning outcomes of similar study programmes offered by reference institutions in the European Higher Education Area:

The learning objectives of the analogous study cycles are broadly similar, focusing on providing the knowledge and skills needed by a conception-oriented engineer in Computer Science and Engineering. To this end, the courses of the core block address core and optional topics from the ACM/IEEE curriculum, with students having to enroll in courses of various areas, for a perspective of breadth. The nature of the specialization tier varies: there are programmes that enforce a focus in a particular sub-field; others, like our proposal, allow the student to choose between concentrating in a sub-field or to flexibly pick themes, in a perspective of broadening his education.

Some programmes (e.g. EPFL) include a mandatory internship of between 8 weeks and 6 months, which may be combined with the master's dissertation. In the proposed programme, there is the possibility of completing the degree with an Engineering Project with similar objectives.

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._Entidades_MEI.pdf](#)

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).

<sem resposta>

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:

n/a

11.3. Institution's own resources to effectively follow its students during the in-service training periods:

n/a

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).

<sem resposta>

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.

Currículo constantemente atualizado, de banda larga, compatível com as recomendações ACM/IEEE 2013, permitindo um percurso flexível e alinhado com a investigação, desenvolvimento tecnológico, e mercado atual. Cobertura de competências transversais pelo "Perfil Curricular FCT".

Todos os docentes doutorados e membros do centro de investigação NOVA LINCS, classificado com excelente nas duas últimas avaliações da FCT. Integração em redes de investigação e educação avançada, nacionais e internacionais. Programas do DI e FCT NOVA para o envolvimento dos alunos em atividades de investigação.

Elevada empregabilidade. Boa ligação com empresas, facilitando o envolvimento dos alunos em Dissertações em ambiente empresarial (Projetos de Engenharia e Dissertações Científicas). Excelente apreciação pelos empregadores e pelos alunos da formação do DI.

Excelentes infraestruturas de ensino e investigação.

12.1. Strengths:

Experience of more than 40 years of DI FCT NOVA in Computer Science and Engineering higher education.

Modern broadband programme, compliant with ACM/IEEE 2013, allowing for a flexible path and aligned with current research, technological development, and market requirements. Coverage of transversal competences by the "FCT Profile".

All faculty with a PhD degree and members of NOVA LINCS research centre, classified as excellent in the last two FCT assessments. Integration in national and international research and advanced education networks. DI and FCT NOVA programs to engage students in research activities.

High employment rate for conception-oriented engineers in CSE. Good connections with firms and industry, which facilitates the involvement of students in dissertations in an industrial setting (both Engineering Projects and Scientific Dissertations). Excellent appreciation of DI FCT education by employers and former students.

Excellent teaching and research infrastructures.

12.2. Pontos fracos:

Dificuldade em reforçar e renovar os recursos humanos e materiais disponíveis no passado recente contribuindo para um aumento da idade média do corpo docente.

Desequilíbrio de género. Apesar das iniciativas do DI, da FCT NOVA e de outras entidades públicas e privadas, o número de candidatas do sexo feminino continua a ser menos de 15%, e do corpo docente de 21%.

Seria desejável atrair mais alunos oriundos de cursos congéneres de outras escolas, para aumentar a diversidade.

Seria positivo desenvolver melhores processos de integração de alunos provenientes de outras áreas científicas, o que também requereria mais recursos.

12.2. Weaknesses:

Difficulty in strengthening and renewing the human and material resources available, in the recent past, has contributed to an increase in the average age of faculty.

Gender imbalance. Despite the initiatives of DI, FCT NOVA and other public and private entities, the number of female candidates remains less than 15% of the total. The percentage of female faculty is 21%.

It would be desirable to attract more students from similar degrees of other schools to increase diversity.

It would be good to develop better processes to integrate students with 1st cycle degrees in other scientific areas. Addressing this challenge would require more resources.

12.3. Oportunidades:

Crescente necessidade de Engenheiros Informáticos de Conceção no momento atual, com um currículo atualizado e alinhado com a investigação, desenvolvimento tecnológico, e mercado atual.

Organização do curso, com percurso flexível, Projeto de Engenharia em contexto empresarial ou Dissertação em ambiente académico-empresarial, pode permitir uma ainda maior colaboração com empresas privadas e instituições públicas.

Oportunidade para atrair estudantes de elevado potencial que tenham completado o 1.º ciclo noutra escola, em Portugal ou noutro país.

O envolvimento dos estudantes em atividades de investigação durante a Dissertação, permite reforçar a base de recrutamento interna de doutorandos.

Oportunidade para a criação de novas empresas de cariz tecnológico, em resultado da combinação das competências adquiridas na área do empreendedorismo, no incentivo à inovação (e.g. nas UC de especialização e na Dissertação), e o crescente suporte na escola e no país a estas iniciativas.

12.3. Opportunities:

Growing need of conception-oriented engineers in CSE, with a modern curriculum and aligned with current research, technological development, and market requirements.

Programme structure, with flexible path including an Engineering Project in industrial context or a Dissertation in academic/industrial context, can foster new and closer collaboration with private companies and public institutions.

Opportunity to attract high-potential students who have completed 1st cycle programmes in other schools, in Portugal or abroad.

The involvement of students in research activities during the dissertation reinforces the internal recruitment base for the PhD programme.

Opportunity for the creation of new technology companies, as a result of the combination of skills acquired in the area of entrepreneurship, the encouragement of innovation (e.g. in the specialization courses and Dissertation), and the growing support at school and in the country for these initiatives.

12.4. Constrangimentos:

Apesar das melhorias recentes, ainda não foi possível recuperar o atraso no reforço e renovação dos recursos humanos e materiais afetos ao curso, o que poderá limitar o desenvolvimento de novas atividades e a atualização das existentes.

A pressão da oferta do mercado de trabalho nesta área, poderá levar mais estudantes a decidirem ingressar prematuramente no mercado de trabalho no fim do 1.º ciclo, desvalorizando a formação necessária a um Engenheiro Informático de Conceção habilitado a maior geração de valor.

A necessidade de profissionais na área da Informática poderá criar pressões para o aumento do número de alunos dos cursos, levando à sua massificação, sem os necessários recursos adicionais.

O surgimento de cursos de banda estreita em áreas de elevada visibilidade (e.g. Inteligência Artificial) pode diminuir a atratividade dos cursos de banda larga. No curso agora proposto, a possibilidade do aluno concentrar os seus estudos numa área, permite atenuar esse constrangimento.

12.4. Threats:

Despite recent improvements, it has not yet been possible to catch up on strengthening and renewing the human and material resources allocated to the programme, which may limit the development of new activities and the updating of existing ones.

The pressure from the industry in this area may lead to more students deciding to enter the labour market prematurely at the end of the 1st cycle, devaluing the education required for conception-oriented engineering, which is able to generate greater value.

The need for IT professionals could create pressure to increase the number of students in the programmes, leading to their massification, without the necessary additional resources.

The emergence of narrowband programmes in high visibility areas (e.g. Artificial Intelligence) may diminish the attractiveness of broadband programmes. In the programme now proposed, the possibility of the student concentrating his studies in one area, allows to mitigate this threat.

12.5. Conclusões:

A presente proposta de curso de Mestrado em Engenharia Informática consubstancia uma visão atualizada sobre a formação do Engenheiro Informático de conceção (nível E2 de qualificação da Ordem dos Engenheiros e acreditação EUR-ACE).

O objetivo do curso é formar Engenheiros Informáticos de conceção habilitados a desenvolver atividades de projeto, liderança, e inovação, por vezes em contexto de investigação, e com bases para aceder a um 3.º Ciclo, completando a formação de base obtida num curso de 1.º ciclo em Engenharia Informática. Estes objetivos estão alinhados com o projeto educativo, científico e cultural da Escola.

A estrutura do curso está organizada pressupondo dos candidatos uma formação de base compatível com as recomendações do ACM/IEEE CS Curriculum 2013 para um primeiro ciclo, e visa consolidar e aprofundar as competências, orientadas para a inovação, para a conceção de serviços e produtos, e para investigação científica. A estrutura do curso permite um percurso flexível, adaptado aos interesses dos estudantes, com o bloco base fornecendo uma formação sólida, de banda larga, complementada por um bloco de especialização que permite ao estudante aprofundar conhecimentos numa subárea da Engenharia Informática ou aprofundar os seus conhecimentos num conjunto de temas do seu interesse. O curso culmina com a Preparação de Dissertação/Projeto e uma Dissertação científica a realizar em ambiente académico ou académico-empresarial ou num Projeto de Engenharia a realizar em ambiente empresarial.

O plano curricular está concebido de forma a expor todos os estudantes à inovação, potenciando o seu envolvimento em atividades de investigação, reforçando a base de recrutamento interna de doutorandos. O plano curricular fomenta ainda a autonomia dos estudantes, com o objetivo de preparar profissionais capazes de evoluírem com sucesso numa área intensamente sujeita à mudança, como é o caso da Informática. A adequação ao “Perfil Curricular FCT”, comum a todos os cursos da escola, favorece o desenvolvimento de várias competências transversais, em particular ligadas ao empreendedorismo e aos aspetos éticos e sócio-profissionais da Informática.

A empregabilidade dos cursos de Engenharia Informática da FCT NOVA tem sido superior a 98%. Acredita-se que a proposta aqui apresentada oferece uma resposta eficaz, atualizada e oportuna, à sustentada necessidade de profissionais cada vez mais qualificados em Engenharia Informática, sendo compatível com as mais elevadas referências de formação afim na Europa e Estados Unidos.

12.5. Conclusions:

The present proposal for the Master programme in Computer Science and Engineering (CSE) implements a modern view on the education of conception-oriented Computer Science/Informatics Engineers (E2 level of qualification of the Ordem dos Engenheiros and EUR-ACE accreditation).

The aim of the programme is to educate conception-oriented Computer Science Engineers prepared to develop project, leadership, and innovation activities, sometimes in an R&D context, with the basis to pursue a 3rd Cycle, by completing the basic education obtained in a 1st cycle in CSE. These objectives are aligned with the school's educational, scientific and cultural project.

The programme structure is organized on the assumption that the candidates have a base education compatible with the recommendations of the ACM / IEEE CS Curriculum 2013 for a first cycle, and aims to consolidate and deepen innovation-oriented skills for the design of services and products, and for scientific research. The programme structure allows for a flexible academic record tailored to the students' interests, with the core block providing solid, broad education complemented by a specialization block that allows students to deepen knowledge in a sub-area of CSE or to deepen their knowledge in a set of topics of interest. The programme concludes with the Dissertation/Project Preparation and a scientific Dissertation to be done in an academic or academic-industrial setting or an Engineering Project to be done in an industrial setting.

The programme is designed to expose all students to innovation, fostering their involvement in research activities, and strengthening the internal recruitment base of doctoral students. The programme also promotes student autonomy,

with the objective of preparing professionals to successfully evolve their knowledge and skills, which is fundamental in an area intensely subject to change as it is the case of CSE. The “FCT Curricular Profile”, common to all school programmes, favors the development of various transversal competences, in particular related to entrepreneurship and the ethical and socio-professional aspects of Computer Science.

The employment rate of FCT NOVA’s Computer Science and Engineering programmes has been over 98%. The proposal presented here provides an effective, modern and timely response to the sustained need for increasingly qualified Computer Science and Engineering professionals, being in line with the programmes offered in reference schools in Europe and in the United States.