

# NCE/19/1901164 — Apresentação do pedido - Novo ciclo de estudos

---

## 1. Caracterização geral do ciclo de estudos

### 1.1. Instituição de Ensino Superior:

*Universidade Do Porto*

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

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

*Faculdade De Engenharia (UP)*

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

*Faculdade De Ciências (UP)*

### 1.3. Designação do ciclo de estudos:

*Engenharia Informática e Computação*

### 1.3. Study programme:

*Informatics and Computing Engineering*

### 1.4. Grau:

*Licenciado*

### 1.5. Área científica predominante do ciclo de estudos:

*Engenharia Informática e Computação*

### 1.5. Main scientific area of the study programme:

*Informatics and Computing 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:

*529*

### 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:

*481*

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

*180*

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

*6 semestres*

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

*6 semesters*

### 1.9. Número máximo de admissões:

*410*

**1.10. Condições específicas de ingresso.**

*Para acesso à Licenciatura em Engenharia Informática e Computação (LiEIC), através do Concurso Nacional de Acesso é necessário:*

- Ser titular de um curso de ensino secundário, ou de habilitação legalmente equivalente;
- Ter realizado os exames nacionais correspondentes às provas de ingresso exigidas para o ciclo de estudos, mais concretamente, um dos seguintes conjuntos de provas: Física e Química + Matemática A ou Português + Matemática A;
- Ter obtido em cada uma das provas de ingresso a classificação mínima fixada;
- Ter obtido, na nota de candidatura, a classificação mínima fixada.

*Verificam-se ainda ingressos no ciclo de estudos através dos seguintes concursos: Regimes Especiais, Concurso Especial de Acesso e Ingresso para Estudantes Internacionais, Reingresso, Mudança de Par Instituição/Curso, Concursos Especiais (Maiores de 23, CET-Cursos de Especialização Tecnológica, Titulares de Outros Cursos Superiores, Titulares de Diploma de Curso Técnico Superior Profissional).*

**1.10. Specific entry requirements.**

*In order to be admitted to this study cycle through the National Access regime, applicants must:*

- Hold a secondary education course, or legal equivalent;
- Attend the mandatory secondary school examinations, specifically one of the following sets of exams: Physics and Chemistry + Mathematics A or Portuguese + Mathematics A;
- Obtain in each of the admission exams the minimum required grade;
- Obtain the minimum required classification in the application grade.

*There are also admissions in the study cycle in the following ways: Special Regimes, Special Access and Admission Contest for International Students, Re-entry, Change of Institution / Course Pair, Special Regimes (Over 23, CET-Technological Specialization Courses, Holders of Other Higher Education Degrees, Holders of Higher Professional Technical Course Diploma).*

**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:**

*Faculdade de Engenharia da Universidade do Porto*

**1.12. Premises where the study programme will be lectured:**

*Faculty of Engineering of the University of Porto*

**1.13. Regulamento de creditação de formação académica e de experiência profissional, publicado em Diário da República (PDF, máx. 500kB):**

[1.13.\\_Regulamento\\_no\\_42\\_2019\(1\).pdf](#)

**1.14. Observações:**

*Este ciclo de estudos resulta da conjugação de esforços da FEUP e FCUP que, no âmbito do processo de reestruturação do Mestrado Integrado em Engenharia Informática e Computação (MIEIC) da FEUP e do Mestrado Integrado em Engenharia de Redes e Sistemas Informáticos (MIERSI) da FCUP, de forma a dar cumprimento ao artigo 19.º do Decreto-Lei n.º 65/2018 de 16 de agosto, optaram por propor um primeiro ciclo único conjunto de banda larga em Engenharia Informática e Computação – a presente Licenciatura em Engenharia Informática e Computação (LiEIC). Tendo em conta o sucesso e reconhecimento no mercado do MIEIC, a oferta formativa de excelência do MIERSI, a constante evolução tecnológica e científica na área de Engenharia Informática e Computação e a constante evolução das práticas e tecnologias pedagógicas, aproveitaram-se diversos pontos fortes do MIEIC e do MIERSI e introduziram-se diversas atualizações e melhorias.*

*A conjugação de esforços entre a FEUP e a FCUP permitiu realizar esta proposta, com o objetivo de reforçar a oferta de formação superior em Engenharia Informática na Universidade do Porto, aproveitando as valências complementares das duas unidades orgânicas.*

*A LiEIC está estruturada em 6 semestres, totalizando 180 ECTS, dos quais 175.5 são obrigatórios e 4.5 são optativos, culminando num Projeto Integrador que decorre no último semestre em paralelo com outras unidades curriculares (UCs).*

*O número máximo de admissões proposto para a LiEIC (410) justifica-se pela junção das vagas e dos corpos docentes dos anteriores mestrados integrados da FEUP e FCUP que lhe deram origem (MIEIC e MIERSI), bem como pelo facto de os números de vagas atuais já estarem muito próximo do máximo de admissões e com tendência para aumentar (devido nomeadamente ao índice de excelência e à procura crescente por estudantes internacionais).*

**1.14. Observations:**

*This study cycle results from the joint efforts of FEUP and FCUP that, in the process of restructuring their Integrated Master in Informatics and Computing Engineering (MIEIC/FEUP) and Integrated Master in Network and Information*

*Systems Engineering (MIERSI/FCUP), in order to comply with article 19 of Decree-Law no. 65/2018 of 16 August, decided to propose a joint broadband first cycle in Informatics and Computing Engineering – the current “Licenciatura” in Informatics and Computing Engineering (LiEIC).*

*Taking into account the success and market recognition of MIEIC, MIERSI's excellent training offer, the constant technological and scientific evolution in the area of Informatics Engineering and Computing, and the constant evolution of pedagogical practices and technologies, several strengths of MIEIC and MIERSI were taken advantage of and several updates and improvements were introduced.*

*The combination of efforts between FEUP and FCUP made it possible to carry out this proposal, with the aim of reinforcing the offer of higher education in Informatics Engineering at the University of Porto, taking advantage of the complementary competences of the two organic units.*

*The study cycle is structured with 180 ECTS, of which 175.5 are compulsory and 4.5 are optional, culminating in a Capstone Project that takes place in the last semester, in parallel with other curricular units (UCs).*

*The maximum number of admissions proposed for LiEIC (410) is justified by the combination of vacancies and teaching staff of the previous integrated masters of FEUP and FCUP (MIEIC and MIERSI), as well as the fact that the current numbers of vacancies are already very close to the maximum number of admissions allowed and with a tendency to increase (namely due to the excellence index and the growing demand by international students).*

## 2. Formalização do Pedido

### Mapa I - Conselho Científico da Faculdade de Engenharia da Universidade do Porto

---

#### 2.1.1. Órgão ouvido:

*Conselho Científico da Faculdade de Engenharia da Universidade do Porto*

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

[2.1.2.\\_Oficio\\_Criação CC LiEIC FEUP 1.pdf](#)

### Mapa I - Conselho Científico da Faculdade de Ciências da Universidade do Porto

---

#### 2.1.1. Órgão ouvido:

*Conselho Científico da Faculdade de Ciências da Universidade do Porto*

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

[2.1.2.\\_Parecer\\_CC\\_FCUP\\_NCE\\_LiEIC.pdf](#)

### Mapa I - Conselho Pedagógico da Faculdade de Engenharia da Universidade do Porto

---

#### 2.1.1. Órgão ouvido:

*Conselho Pedagógico da Faculdade de Engenharia da Universidade do Porto*

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

[2.1.2.\\_Oficio\\_Criação CP LiEIC FEUP 1.pdf](#)

### Mapa I - Conselho Pedagógico da Faculdade de Ciências da Universidade do Porto

---

#### 2.1.1. Órgão ouvido:

*Conselho Pedagógico da Faculdade de Ciências da Universidade do Porto*

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

[2.1.2.\\_Parecer\\_CP\\_FCUP\\_NCE\\_LiEIC.pdf](#)

### Mapa I - Reitor da Universidade do Porto

---

#### 2.1.1. Órgão ouvido:

*Reitor da Universidade do Porto*

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

[2.1.2.\\_Despacho reitoral 1º Eng Informatica e Computação.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 geral da LiEIC é promover a excelência da qualificação inicial nos fundamentos científicos e técnicos da Engenharia Informática e Computação (EIC), preparando os diplomados para o aprofundamento de estudos numa formação de 2º ciclo e subsequente desempenho de funções de grande responsabilidade em contextos complexos e com foco na inovação, ou para a entrada no mercado de trabalho para desempenho de funções compatíveis com uma formação de 3 anos em EIC.*

*Para isso, oferece uma formação conciliando:*

- *uma formação cuidada nas ciências básicas de Engenharia, nomeadamente na Matemática e Física;*
- *uma formação de base abrangente nas grandes áreas científicas e técnicas de EIC, alinhada com as boas práticas e recomendações internacionais, nomeadamente o “Computer Science Curriculum 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science” do ACM e IEEE;*
- *uma formação flexível em competências transversais, nomeadamente “soft skills”.*

### **3.1. The study programme’s generic objectives:**

*LiEIC’s general objective is to promote the excellence of the initial qualification in the scientific and technical foundations of Informatics Engineering and Computing (EIC), preparing graduates for further studies in a 2nd cycle and subsequent undertaking of functions of high responsibility in complex contexts with a focus on innovation, or entering the job market to perform jobs compatible with a 3-year background in IEC.*

*To this end, it offers a training combining:*

- *a thorough education in the basic engineering sciences, namely in Mathematics and Physics;*
- *comprehensive background training in the major scientific and technical areas of EIC, aligned with international best practices and recommendations, notably the “Computer Science Curriculum 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science” from the ACM and IEEE Computer Society;*
- *flexible training in transversal skills, including soft skills.*

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

*Com referência aos descritores CDIO, e para problemas e sistemas baseados em TIC de média complexidade, os objetivos são:*

- *Adquirir com a necessária proficiência conhecimentos de*
  - *ciências básicas (1.1), como Matemática e Física,*
  - *ciências de engenharia (1.2) e tecnologias da sua área de formação (1.3)*
- e ser capaz de os utilizar*
  - *na formulação, resolução e discussão de problemas na sua área de formação;*
  - *na conceção de soluções para problemas e na antecipação e prevenção dos mesmos;*
  - *no projeto, implementação e operação de sistemas (1).*
- *Adquirir com a necessária proficiência capacidades e atitudes pessoais e profissionais (2) e interpessoais (3), incluindo: raciocínio em engenharia e resolução de problemas; experimentação e descoberta do conhecimento; pensamento sistémico; trabalho em grupo; comunicação.*
- *Ser capaz de conceber, projetar, implementar e operar sistemas no contexto social e empresarial (4).*

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

*With reference to CDIO descriptors, and for medium complexity ICT based problems and systems, the expected learning outcomes are:*

- *To acquire with the necessary proficiency knowledge of*
  - *basic sciences (1.1), such as mathematics and physics,*
  - *engineering sciences (1.2) and technologies in their field of study (1.3)*
- and be able to use them*
  - *in formulating, solving and discussing problems in their area of training;*
  - *in conceiving solutions and anticipating and preventing problems;*
  - *in the design, implementation and operation of systems (1).*
- *Acquire with the necessary proficiency personal and professional (2) and interpersonal (3) skills and attitudes, including: engineering thinking and problem solving; experimentation and knowledge discovery; systemic thinking; group work; communication.*
- *Be able to conceive, design, implement and operate systems in the social and business context (4).*

### **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 LiEIC, ao envolver docentes da FEUP e da FCUP, enquadra-se nas orientações estratégicas da UP de cooperação entre unidades orgânicas com competências complementares, no sentido de fortalecer a oferta da UP na área da Engenharia Informática e Computação.*

*A LiEIC enquadra-se na missão institucional de educação e formação de profissionais de Engenharia de nível internacional, sustentadas em Investigação e Desenvolvimento de excelência com foco regional mas com qualidade reconhecida ao nível global e numa prática de extensão de reconhecida qualidade, contemplando as vertentes científica, técnica, ética e cultural.*

*A LiEIC revê-se completamente nesta missão, ao fornecer uma formação nos fundamentos de Engenharia Informática que prepara os seus graduados para a prossecução de estudos a nível de 2º ciclo, capacitando-os assim para enveredar por carreiras profissionais em termos nacionais e internacionais, como tem ocorrido com inúmeros dos seus graduados. Para isso, confere aos seus estudantes competências variadas, técnicas e pessoais, de importância reconhecida no mundo empresarial.*

*Seguem-se alguns dos objetivos institucionais e comentários de enquadramento da LiEIC.*

- *Formar profissionais de engenharia de reconhecida qualidade e competência.*
- *LiEIC: promover a excelência da qualificação inicial dos seus graduados.*

- *Alcançar uma posição cimeira, tanto a nível nacional como internacional, em atividades de I&D, nomeadamente como suporte de um ensino de qualidade.*
- *LiEIC: conciliar a formação nas ciências da engenharia com a formação em áreas científicas relacionadas com a Engenharia Informática, reforçando uma perspetiva de I&D através da relação dos seus docentes com laboratórios associados e institutos de investigação de topo.*
- *Assegurar o permanente desenvolvimento dos recursos computacionais e de informação, nomeadamente nos aspetos de capacidade, qualidade e acessibilidade.*
- *LiEIC: pela sua natureza, o ciclo de estudos é um utilizador privilegiado destes recursos, colaborando, pontualmente, no seu desenvolvimento, em trabalhos realizados pelos estudantes.*
- *Incrementar o seu relacionamento com a comunidade envolvente.*
- *Incrementar as ações de cooperação com Instituições de reconhecido prestígio internacional, tanto ao nível do ensino como das atividades de I&D.*
- *LiEIC: mantém relações com empresas e institutos nacionais de I&D, nomeadamente ao nível da colaboração dos seus docentes e de projetos extracurriculares desenvolvidos por estudantes.*
- *Investir na criação de uma imagem forte, consistente e apelativa que desempenhe um importante papel na criação do espírito de escola, estimule o interesse dos potenciais estudantes e contribua para a sua afirmação na comunidade.*
- *LiEIC: postura de divulgação ao grande público, com ênfase no ensino secundário, com os eventos "Semana Profissão Engenheiro", "Dias Abertos", "TOPAS", "Mostra da UP", receção de visitantes com visitas/demonstrações laboratoriais, visitas a escolas secundárias, etc.*

### **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:**

*LiEIC, by involving faculty members from FEUP and FCUP, fits into the strategic guidelines of the UP for cooperation between organic units with complementary competences, in order to strengthen the UP offer in the area of Informatics and Computing Engineering.*

*LiEIC fits into the institutional mission of education and training of world-class Engineering professionals, supported by excellence in Research and Development with a regional focus but with recognized quality at a global level and in an extension practice of recognized quality, covering the scientific, technical, ethical and cultural aspects.*

*LiEIC fully revises this mission by providing training in the foundations of Informatics Engineering that prepares its graduates for further studies at the 2nd cycle level, thus enabling them to pursue professional careers at national and international level, such as has occurred with numerous graduates. To this end, it gives its students varied technical and personal skills of recognized importance in the business world.*

*Here are some of institutional objectives and explanations about how LiEIC fulfils those objectives.*

- *Train engineering professionals of recognized quality and competence.*
- *LiEIC: Promote excellence in the initial qualification of its graduates.*
- *Achieving a top position, both nationally and internationally, in R&D, notably in support of quality education.*
- *LiEIC: reconcile engineering sciences training with computer science training, reinforcing an R&D perspective through the collaboration of its faculty with laboratories and top research institutes.*
- *To ensure the permanent development of computational and information resources, namely in terms of capacity, quality and accessibility.*
- *LiEIC: By its very nature, the study cycle is a privileged user of these resources, collaborating, occasionally, in its development, in works done by students.*
- *Increase its relationship with the surrounding community.*
- *Increase cooperation with institutions of recognized international prestige, both in education and in R&D activities.*
- *LiEIC: maintains relationships with companies and national R&D institutes in various aspects, namely in terms of collaboration of their teachers and extracurricular projects developed by students.*
- *Invest in creating a strong, consistent and appealing image that plays an important role in creating a school spirit, stimulating the interest of potential students and contributing to their affirmation in the community.*
- *LiEIC: public outreach, with emphasis on secondary education, with the events "Semana Profissão Engenheiro", "Dias Abertos", "TOPAS", "Mostra da UP", reception of visitors with visits/laboratory demonstrations, visits to secondary schools, etc.*

## **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:

N/A

N/A

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

**Mapa II - N/A****4.2.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):**

N/A

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

N/A

**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
Engenharia Informática e Computação / Informatics Engineering and Computing	EINFC	136.5	0	
Física / Physics	FIS	10.5	0	
Matemática / Mathematics	M	27	0	
Desenvolvimento Pessoal / Personal Development	DP	1.5	0	
Desenvolvimento Pessoal/ Competências transversais/transferíveis/Qualquer área científica da Universidade do Porto (UP)	DP/CTT/QACUP	0	4.5	
<b>(5 Items)</b>		<b>175.5</b>	<b>4.5</b>	

**4.3 Plano de estudos****Mapa III - N/A - 1º Ano / 1º Semestre - 1st Year / 1st Semester****4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):**

N/A

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

N/A

**4.3.2. Ano/semestre/trimestre curricular:**

1º Ano / 1º Semestre - 1st Year / 1st Semester

**4.3.3 Plano de Estudos / Study plan**

Unidade Curricular / Curricular Unit	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS / Observations (5)
Álgebra Linear e Geometria Analítica / Linear Algebra and Analytical Geometry	M	Semestral	121.5	T: 19.5; TP: 19.5	4.5
Análise Matemática I / Mathematical Analysis I	M	Semestral	162	T: 26; TP: 26	6
Fundamentos da Programação / Programming Fundamentals	EINFC	Semestral	162	T: 26; TP: 26	6
Fundamentos de Sistemas Computacionais / Fundamentals of Computer Systems	EINFC	Semestral	162	T: 26; TP: 26	6
Matemática Discreta / Discrete Mathematics	M	Semestral	162	T: 26; TP: 26	6
Projeto UP / Project UP	DP	Semestral	40.5	T: 3; TP: 10	1.5 a)
<b>(6 Items)</b>					

**Mapa III - N/A - 1º Ano / 2º Semestre - 1st Year / 2nd Semester****4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):**

N/A

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

N/A

**4.3.2. Ano/semestre/trimestre curricular:***1º Ano / 2º Semestre - 1st Year / 2nd Semester***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)
Arquitetura de Computadores / Computer Architecture	EINFC	Semestral	162	T: 26; TP: 26	6	
Análise Matemática II / Mathematical Analysis II	M	Semestral	162	T: 26; TP: 26	6	
Física I / Physics I	FIS	Semestral	162	T: 26; TP: 26	6	
Programação / Programming	EINFC	Semestral	162	T: 26; TP: 26	6	
Teoria da Computação / Theory of Computation	EINFC	Semestral	162	T: 26; TP: 26	6	

(5 Items)

**Mapa III - N/A - 2º Ano / 1º Semestre - 2nd Year / 1st Semester****4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):***N/A***4.3.1. Branch, option, profile, major/minor or other (if applicable):***N/A***4.3.2. Ano/semestre/trimestre curricular:***2º Ano / 1º Semestre - 2nd Year / 1st Semester***4.3.3 Plano de Estudos / Study plan**

Unidade Curricular / Curricular Unit	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS	Observações / Observations (5)
Algoritmos e Estruturas de Dados / Algorithms and Data Structures	EINFC	Semestral	162	T: 26; TP: 26	6	
Bases de Dados / Databases	EINFC	Semestral	162	T: 26; TP: 26	6	
Física II / Physics II	FIS	Semestral	121.5	T: 19.5; TP: 19.5	4.5	
Laboratório de Desenho e Teste de Software / Software Design and Testing Laboratory	EINFC	Semestral	162	T: 26; PL: 26	6	
Sistemas Operativos / Operating Systems	EINFC	Semestral	162	T: 26; TP: 26	6	
Competências Transversais I / Transferable Skills I	DP/CTT/QACUP	Semestral	40.5	depende da UC selecionada	1.5	Optativa, b) Para efeitos de cálculo, estimamos 13 horas de contacto

(6 Items)

**Mapa III - N/A - 2º Ano / 2º Semestre - 2nd Year / 2nd Semester****4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):***N/A***4.3.1. Branch, option, profile, major/minor or other (if applicable):***N/A***4.3.2. Ano/semestre/trimestre curricular:***2º Ano / 2º Semestre - 2nd Year / 2nd Semester***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)
Desenho de Algoritmos / Algorithm Design	EINFC	Semestral	162	T: 26; TP: 26	6	
Engenharia de Software / Software Engineering	EINFC	Semestral	162	T: 26; TP: 26	6	
Laboratório de Computadores / Computer Laboratory	EINFC	Semestral	162	T: 26; PL: 26	6	
Linguagens e Tecnologias Web / Web Languages and Technologies	EINFC	Semestral	162	T: 26; TP: 26	6	
Métodos Estatísticos / Statistical Methods	M	Semestral	121.5	T: 19.5; TP: 19.5	4.5	
Competências Transversais II / Transferable Skills II	DP/CTT/ QACUP	Semestral	40.5	depende da UC selecionada	1.5	Optativa (b) Para efeitos de cálculo, estimamos 13 horas de contacto

(6 Items)

**Mapa III - N/A - 3º Ano / 1º Semestre - 3rd Year / 1st Semester****4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):**

N/A

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

N/A

**4.3.2. Ano/semestre/trimestre curricular:**

3º Ano / 1º Semestre - 3rd Year / 1st Semester

**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)
Fundamentos de Segurança Informática / Computer Security Foundations	EINFC	Semestral	162	T: 26; TP: 26	6	
Interação Pessoa Computador / Human Computer Interaction	EINFC	Semestral	121.5	T: 19.5; TP: 19.5	4.5	
Laboratório de Bases de Dados e Aplicações Web / Database and Web Applications Laboratory	EINFC	Semestral	162	T: 26; PL: 26	6	
Programação Funcional e em Lógica / Functional and Logic Programming	EINFC	Semestral	162	T: 26; TP: 26	6	
Redes de Computadores / Computer Networks	EINFC	Semestral	162	T: 26; TP: 26	6	
Competências Transversais III / Transferable Skills III	DP/CTT/ QACUP	Semestral	40.5	depende da UC selecionada	1.5	Optativa (b) Para efeitos de cálculo, estimamos 13 horas de contacto

(6 Items)

**Mapa III - N/A - 3º Ano / 2º Semestre - 3rd Year / 2nd Semester****4.3.1. Ramo, opção, perfil, maior/menor ou outra (se aplicável):**

N/A

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

N/A

**4.3.2. Ano/semestre/trimestre curricular:**

3º Ano / 2º Semestre - 3rd Year / 2nd Semester

**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)
Compiladores / Compilers	EINFC	Semestral	162	T: 26; TP: 26	6	
Computação Gráfica / Computer Graphics	EINFC	Semestral	162	T: 26; TP: 26	6	
Computação Paralela e Distribuída / Parallel and Distributed Computing	EINFC	Semestral	162	T: 26; TP: 26	6	
Inteligência Artificial / Artificial Intelligence	EINFC	Semestral	162	T: 26; TP: 26	6	
Projeto Integrador / Capstone Project (5 Items)	EINFC	Semestral	162	OT: 13	6	(c)

#### 4.4. Unidades Curriculares

---

##### Mapa IV - Algoritmos e Estruturas de Dados

###### 4.4.1.1. Designação da unidade curricular:

*Algoritmos e Estruturas de Dados*

###### 4.4.1.1. Title of curricular unit:

*Algorithms and Data Structures*

###### 4.4.1.2. Sigla da área científica em que se insere:

*EINFC*

###### 4.4.1.3. Duração:

*Semestral*

###### 4.4.1.4. Horas de trabalho:

*162*

###### 4.4.1.5. Horas de contacto:

*52 (26T + 26TP)*

###### 4.4.1.6. ECTS:

*6*

###### 4.4.1.7. Observações:

*<sem resposta>*

###### 4.4.1.7. Observations:

*<no answer>*

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

*Pedro Manuel Pinto Ribeiro (T-26h – 2 turmas; TP – 52h – 11 turmas)*

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

*Ana Paula Cunha da Rocha (T-26h – 2 turmas; TP-52h - 11 turmas)*

*Bruno Serra Loff Barreto (TP – 52h – 11 turmas)*

*Rosaldo José Fernandes Rossetti (TP-26h - 11 turmas)*

*Docente(s) a contratar (TP- 104h - 11 turmas)*

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

*No final da unidade curricular, os estudantes deverão ser capazes de:*

- analisar (e medir experimentalmente) a complexidade temporal e espacial de algoritmos;*
- analisar (e testar experimentalmente) a correção de algoritmos simples com contratos e lógica de Hoare;*
- conhecer os principais algoritmos de pesquisa e ordenação em arrays e a sua complexidade;*
- compreender o conceito de tipo abstrato de dados e saber organizar programas em torno desse conceito;*
- conhecer as estruturas de dados fundamentais (e algoritmos associados e respetiva complexidade) usadas para implementar eficientemente tipos abstratos de dados comuns em bibliotecas de coleções;*
- saber escolher coleções, estruturas de dados e algoritmos apropriados para resolver problemas práticos;*
- escrever programas em C++ que implementam e usam as estruturas de dados e algoritmos fundamentais.*

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

*At the end of the course, students should be able to:*

- a) analyze (and measure experimentally) the temporal and spatial complexity of algorithms;*
- b) analyze (and test experimentally) the correctness of simple algorithms using contracts and Hoare logic;*
- c) know the main array search and sorting algorithms and their complexity;*
- d) understand the concept of abstract data type and know how to organize programs around this concept;*
- e) know the fundamental data structures (and associated algorithms and respective complexity) used to efficiently implement common abstract data types in collection libraries;*
- f) choose appropriate collections, data structures and algorithms to solve practical problems;*
- g) write programs in C++ that implement and use the fundamental data structures and algorithms.*

**4.4.5. Conteúdos programáticos:**

- 1) Conceitos e técnicas básicas: complexidade temporal e espacial de algoritmos; tipos abstratos de dados; análise da correção de algoritmos com contratos e lógica de Hoare.*
- 2) Algoritmos de pesquisa e ordenação em arrays.*
- 3) Estruturas de dados lineares e sua implementação: pilhas, filas e listas.*
- 4) Estruturas de dados hierárquicos e sua implementação: árvores binárias; árvores binárias de pesquisa; árvores binárias equilibradas; árvores B.*
- 5) Tabelas de dispersão e algoritmos de manipulação.*
- 6) Filas de prioridade e heaps binários.*
- 7) Algoritmos básicos em grafos: tipos de grafos; representação; pesquisa em largura e em profundidade; ordenação topológica; ciclos; conectividade; caminho mais curto; árvore de expansão mínima.*

**4.4.5. Syllabus:**

- 1) Basic concepts and techniques: space and time complexity of algorithms; abstract data types; analysis of algorithm correctness with contracts and Hoare logic.*
- 2) Arrays searching and sorting algorithms.*
- 3) Linear data structures and their implementation: lists, stacks, and queues.*
- 4) Hierarchical data structures and their implementation: binary trees; binary search trees; balanced binary trees (AVL and splay); B-trees.*
- 5) Hash tables and related algorithms.*
- 6) Priority queues and binary heaps.*
- 7) Basic graph algorithm: types of graphs; representation; depth-first and breadth-first search; topological sorting; cycles; connectivity; shortest path; minimum spanning tree.*

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

*Os conteúdos programáticos estão mapeados para os objetivos de aprendizagem da seguinte forma:*

*1 -> a, b, d*

*2 -> c*

*3, 4, 5, 6, 7 -> e, f*

*O objetivo g) é atingido através de exercícios de programação em C++ desenvolvidos nas aulas práticas e dois pequenos projetos desenvolvidos em grupo fora das aulas.*

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

*The program contents are mapped to the learning objectives as follows:*

*1 -> a, b, d*

*2 -> c*

*3, 4, 5, 6, 7 -> e, f*

*Objective g) is achieved through C++ programming exercises developed in the practical classes, and two small projects developed in teams outside classes.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*As aulas teóricas são usadas para exposição formal da matéria, acompanhada da apresentação de exemplos e sua discussão. Nas aulas práticas os estudantes resolvem exercícios de programação em C++ relacionados com a implementação, análise e aplicação de diversas estruturas de dados e algoritmos associados.*

*Avaliações práticas individuais são realizadas ao longo da u.c., em 3 momentos distintos, onde os estudantes são avaliados a nível teórico e prático, com a realização de um teste escrito e de exercícios individuais em computador, respetivamente (Componente Individual, peso 60%). Os estudantes efetuam trabalhos de grupo de pequena dimensão que são acompanhados fora das aulas e avaliados em dois momentos ao longo da u.c. (Componente de Grupo, peso 40%).*

*Tipo de Avaliação: Avaliação distribuída sem exame final*

*Para aprovação na unidade curricular, o estudante deve obter uma classificação mínima de 40% (8 valores em 20) em qualquer das componentes de avaliação.*

**4.4.7. Teaching methodologies (including students' assessment):**

*Theoretical classes are used for formal presentation of course subjects, accompanied by the presentation of examples and their discussion. In practical classes, students solve programming exercises in C++ related with the implementation, analysis and application of several data structures and associated algorithms. Students also develop a small project outside classes.*

*Individual practical assessments are carried out throughout the UC, in 3 different moments, where students are evaluated at the theoretical and practical level, with an individual written text and individual exercises on the computer, respectively (Individual Component, weight 60%). Students carry out small group work that is accompanied outside classes and assessed in two moments throughout the u.c. (Group component, weight 40%).*

**Assessment Type:** *Distributed assessment without final exam*

*To be approved in the course, the student must obtain a minimum grade of 40% (8 out of 20) in any component of evaluation.*

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

*As aulas teóricas são usadas para exposição formal da matéria, acompanhada da apresentação de exemplos e sua discussão.*

*Nas aulas práticas são resolvidos exercícios de programação em C++, sendo os exercícios focados em temas tratados nas aulas teóricas. As resoluções são geralmente efetuadas em grupos de estudantes, sendo fomentada a discussão sobre as propostas de solução que vão sendo apresentadas.*

*A avaliação é realizada ao longo do semestre, em datas previamente anunciadas, onde os estudantes são continuamente avaliados a nível teórico e prático, a nível individual e de grupo. Ao longo do semestre são realizados três pontos de avaliação individual, que incluem questões de índole teórica e realização de exercícios individuais em computador. Ao longo do semestre são também propostos dois projetos de programação a realizar em grupo, fomentando assim a capacidade de trabalho em equipa. O desenvolvimento destes projetos é feito essencialmente fora das aulas, com apanhamento regular.*

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

*Theoretical classes are for formal exposition of the subjects, with the presentation of examples and their analysis and discussion.*

*In practical classes, programming exercises in C++ are solved, that focus on topics addressed in the theoretical classes. Resolutions are usually in groups of students, and discussion on proposed solutions is encouraged.*

*The assessment is carried out throughout the semester, on previously announced dates, where students are continuously evaluated at the theoretical and practical level, at individual and group level. During the semester, there are three individual assessment points, including theoretical questions and individual computer exercises. Throughout the semester, two programming projects are also proposed, that are solved in group of students, thus fostering the ability to work in teams. The development of these projects is essentially done out of class, with regular monitoring.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Weiss, M. A. (2014). Data Structures and Algorithm Analysis in C++ (4th ed). London: Pearson Education Ltd.*

*Sedgewick, R. (1998). Algorithms in C++, parts 1-4: fundamentals, data structure, sorting, searching (Vol. 1). Pearson Education.*

*Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT press.*

*Stroustrup, B. (2013). The C++ programming language. Pearson Education.*

*Huth, M., & Ryan, M. (2004). Logic in Computer Science. Cambridge University Press. ISBN: 978-0-511-26401-6*

### **Mapa IV - Álgebra Linear e Geometria Analítica**

#### **4.4.1.1. Designação da unidade curricular:**

*Álgebra Linear e Geometria Analítica*

#### **4.4.1.1. Title of curricular unit:**

*Linear Algebra and Analytical Geometry*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*M*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*121.5*

#### **4.4.1.5. Horas de contacto:**

*39 (19.5T + 19.5TP)*

**4.4.1.6. ECTS:**

4.5

**4.4.1.7. Observações:**

*Esta unidade curricular tem dois objetivos fundamentais: por um lado, tratando-se de uma unidade curricular propedêutica tem um carácter didático/científico, promovendo o desenvolvimento do raciocínio lógico e de métodos de análise e, por outro, visa introduzir e desenvolver em termos teóricos um conjunto de conceitos que serão ferramentas essenciais para apoio às unidades curriculares mais específicas da Engenharia.*

**4.4.1.7. Observations:**

*This course has two main objectives: the promotion of logical reasoning and methods of analysis and the introduction and theoretical development of a set of concepts that will be fundamental to support the study of other disciplines along this course of studies.*

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

*José Ferreira Alves (T - 39h – 2 turmas; TP - 78h – 10 turmas)*

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

*Ana Paula da Silva Dias (TP - 117h - 10 turmas)*

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

*A unidade visa desenvolver ferramentas matemáticas usadas em Engenharia.*

*O estudante deverá dominar conceitos de cálculo vetorial em  $R^n$ , especialmente  $R^2$  e  $R^3$ . Deve ser capaz de usar propriedades de matrizes, determinantes, espaços vetoriais e aplicações lineares.*

*Em particular, o estudante deve ser capaz de:*

- reconhecer espaços e subespaços vetoriais reais, determinar suas bases, calcular sua dimensão e coordenadas de vetores numa base;
- reconhecer aplicações lineares e as suas principais propriedades;
- efetuar operações com matrizes e usá-las para resolver sistemas de equações lineares e discuti-los, operar com matrizes associadas a aplicações lineares, determinar vetores e valores próprios, diagonalizar (se possível) uma matriz, e (caso o tempo permita) identificar cônicas e quádricas usando matrizes e valores próprios;
- calcular determinantes, aplicar as suas propriedades e sua interpretação em termos de áreas e volumes.

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

*The unit aims developing mathematical tools used in Engineering.*

*The student should master the main concepts of vector algebra in  $R^n$ , with special emphasis on  $R^2$  and  $R^3$ .*

*Students must be able to use the main properties of the concepts of matrix, determinant, vector space and linear map. In particular, the student should be able to:*

- recognize real vector spaces and subspaces, determine their bases, compute their dimension and coordinates of vectors on a basis;
- recognize linear maps, and their main properties;
- work with matrix operations and use them to solve systems of linear equations and to discuss them, operate with matrices associated with linear maps; determine eigenvectors and eigenvalues, diagonalize a matrix (if possible), and if time permits, identify conics and quadrics using matrices and eigenvalues;
- compute determinants, apply their properties and their geometric interpretation as area and volume.

**4.4.5. Conteúdos programáticos:**

*Sistemas de equações lineares reais e matrizes.*

*Operações com matrizes.*

*Determinante de uma matriz quadrada.*

*O espaço euclidiano  $R^n$ .*

*Aplicações lineares em  $R^n$ .*

*Bases e mudanças de coordenadas.*

*Vetores próprios e valores próprios.*

*(Opcional, caso haja tempo) Cônicas e quádricas.*

**4.4.5. Syllabus:**

*Systems of real linear equations and matrices.*

*Determinant of a square matrix.*

*The Euclidian space  $R^n$ .*

*Linear maps in  $R^n$ .*

*Bases and change of coordinates.*

*Eigenvectors and eigenvalues.*

*(Optional, if time permits) Conics and quadrics.*

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

*O objetivo de dotar o estudante de ferramentas computacionais usadas em Engenharia faz com que a unidade curricular tenha uma forte ênfase computacional.*

*Esta unidade curricular representa um primeiro contato do estudante com a área de álgebra linear em que lhe são introduzidos as definições e resultados básicos referentes à estrutura algébrica de espaço vetorial linear, aplicação linear entre espaços vetoriais lineares e o espaço euclidiano  $R^n$ . Pretende-se que este primeiro contato tenha as duas vertentes algébrica e geométrica presentes para ajudar na compreensão e futuras aplicações do conteúdo programático noutras unidades curriculares. Neste primeiro contato é importante o estudante conhecer um dos métodos de resolução de sistemas lineares, sendo o método de Gauss a escolha para esta unidade, conhecer a notação e operações matriciais, e saber calcular o determinante de uma matriz quadrada usando as propriedades da função determinante.*

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

*The aim of providing students with computational tools used in Engineering implies that this unit has a strong computational emphasis.*

*This curricular unit represents the first contact the student has with the linear algebra area where the basic definitions and results concerning the linear vector space algebraic structure, linear map between linear vector spaces and the euclidian vector space  $R^n$ . Both algebraic and geometric view points are important for the understanding and application of the syllabus in other curricular units. At this first contact, it is important for the student to know at least one of the methods for resolution of systems of linear equations; here the Gauss method is the choice for this unit, to know the matrix notation and operations, and to be familiarized with the computation of matrix determinants using the properties of the determinant function.*

#### **4.4.7. Metodologias de ensino (avaliação incluída):**

*As horas de contacto estão distribuídas em aulas teóricas e teórico-práticas. Nas primeiras são apresentados os conteúdos do programa, recorrendo-se a exemplos para ilustrar os conceitos tratados e orientar os estudantes. Nas aulas teórico-práticas são resolvidos exercícios e problemas, previamente indicados. São disponibilizados materiais de apoio na página da disciplina. Para além das aulas, há períodos de atendimento semanais onde os estudantes têm oportunidade de esclarecer dúvidas.*

*A avaliação será distribuída, com ou sem exame final.*

*A avaliação distribuída poderá incluir a realização de dois ou mais testes escritos, que poderão substituir o exame, se existir.*

*Condições de Frequência: as definidas no artigo 4º das Normas Gerais de Avaliação em vigor na FEUP.*

#### **4.4.7. Teaching methodologies (including students' assessment):**

*Lectures and example classes: The contents of the syllabus are presented in the lectures, where examples are given to illustrate the concepts. There are also practical lessons, giving the solution to previously indicated exercises and problems. All resources are available for students at the unit's web page. The lecturer(s) have weekly office hours for discussion of difficulties with students.*

*Evaluation: continuous evaluation with or without final exam.*

*The continuous evaluation may include two or more written tests, that may replace the final exam, if it exists.*

*Terms of frequency: According to General Evaluation Rules of FEUP.*

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

*As aulas teóricas são organizadas de acordo com o programa proposto, recorrendo-se a exemplos para motivar e exemplificar as diversas partes da matéria, tendo em conta ambas as visões algébrica e geométrica. Os aspetos computacionais são enfatizados nas aulas teórico-práticas. Os estudantes são estimulados a trabalhar com antecedência os problemas propostos para as aulas teórico-práticas, onde são resolvidos e discutidos exercícios com vista a sedimentar os conceitos estudados.*

*Mais especificamente, a resolução dos problemas propostos permitirá aos estudantes atingir os objetivos de aprendizagem de natureza mais aplicada, como "operar com matrizes", "determinar vetores e valores próprios", "calcular determinantes" e "resolver sistemas de equações lineares". O acompanhamento das aulas teóricas e o estudo dos materiais disponibilizados preparará os estudantes para a resolução dos problemas propostos, dotando-os da capacidade de reconhecer e compreender os principais conceitos e técnicas necessários à resolução dos problemas.*

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

*The lectures are organized according to the proposed syllabus and examples are given to motivate and exemplify the different parts of the syllabus, having in mind both the algebraic and geometric view points. Computational aspects are emphasized in example classes. Exercises are proposed in advance before the example classes to motivate the students to work on them earlier on, so that during the example classes, the resolution and discussion of those exercises contribute as well to strengthen the unit concepts that are being studied.*

*More specifically, solving the proposed exercises will allow students to achieve learning objectives of a more applied nature, such as "working with matrices", "determining vectors and eigenvalues", "computing determinants" and "solving systems of linear equations". The participation in the theoretical classes and the study of the available materials will prepare students for the resolution of the proposed exercises, providing them with the ability to recognize and understand the main concepts and techniques necessary for problem solving.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

- Avrizer, D. (2009). *Geometria analítica e álgebra linear: uma visão geométrica*. Editora UFMG. ISBN: 978-85-7041-754-1
- Anton, H., & Rorres, C. (2005). *Elementary linear algebra: applications version*. John Wiley, 2005. ISBN: 0-471-66959-

8<sup>[L]</sup><sub>[SEP]</sub>

- Edwards jr. C.H., & Penney, D.E. (1988). *Elementary linear algebra*. Prentice-hall. ISBN: 0-13-258245-7<sup>[L]</sup><sub>[SEP]</sub>
- Monteiro, A. (2001). *Álgebra linear e geometria analítica*. Mc Graw-Hill. ISBN: 978-9-727-73106-0<sup>[L]</sup><sub>[SEP]</sub>
- Mansfield, L.E. (1976). *Linear algebra with geometric applications*. Marcel Dekker. ISBN: 0-8247-6321-1
- Altland, A., & Delft, J. (2019). *Mathematics for Physicists: Introductory Concepts and Methods*. Cambridge University Press. ISBN-13: 978-1-108-47122-0

#### Mapa IV - Análise Matemática I

##### 4.4.1.1. Designação da unidade curricular:

*Análise Matemática I*

##### 4.4.1.1. Title of curricular unit:

*Mathematical Analysis I*

##### 4.4.1.2. Sigla da área científica em que se insere:

*M*

##### 4.4.1.3. Duração:

*Semestral*

##### 4.4.1.4. Horas de trabalho:

*162*

##### 4.4.1.5. Horas de contacto:

*52 (26T + 26TP)*

##### 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):

*Alexandre Miguel Prior Afonso, T-52h (2 turmas); TP-52h (11 turmas)*

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

*Catarina Rosa Santos Ferreira de Castro, TP-52h (11 turmas)*

*Ana Maria Azevedo Neves, TP-130h (11 turmas)*

*Sónia Isabel Silva Pinto, TP-52h (11 turmas)*

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

*Aquisição de conhecimentos teóricos e práticos sobre cálculo diferencial e integral em R que possibilitem a aplicação das ferramentas básicas da análise matemática ao tratamento e resolução dos problemas adaptados ao perfil da Engenharia Informática e Computação. Capacitar o estudante para a inovação, complementando os conhecimentos de forma a desenvolver soluções para resolução de novas questões.*

*Como resultado da aprendizagem ao longo desta UC, o estudante deve ter adquirido as seguintes competências:*

- 1. Analisar funções, derivar e desenhar gráficos*
- 2. Dominar as técnicas de integração e utilizar os integrais em aplicações de engenharia*
- 3. Compreender e utilizar as equações diferenciais e transformadas de Laplace*
- 4. Saber relacionar séries e polinómios e perceber os conceitos de aproximação.*

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

*This course aims to acquaint students with the differential and integral calculus, in order to make them able to apply basic tools of mathematical analysis in problem solving related to subjects of Informatics and Computing Engineering. This course also aims to expand students' knowledge, so that they can address new methodologies applied to engineering problems. As a result of this course, students should be acquainted with the following matters:*

- 1. To study functions, solve derivatives and draw graphics*
- 2. To solve integrals and use them in various engineering applications*
- 3. To use differential equations and Laplace Transform*
- 4. To understand approximation concepts using series and polynomials.*

**4.4.5. Conteúdos programáticos:**

- 1- *Diferenciação*
  - a. *Aplicações da diferenciação à engenharia*
  - b. *Cálculo de limites*
- 2- *Integração*
  - a. *Integral indefinido*
  - b. *Integral definido*
  - c. *Teorema fundamental do cálculo*
  - d. *Técnicas de integração*
  - e. *Aplicações de Integração – Cálculo de áreas e volumes*
- 3- *Equações Diferenciais*
  - a. *Equações Diferenciais de 1ª ordem*
  - b. *Equações Diferenciais de segunda ordem*
- 4- *Transformadas de Laplace e sua aplicação à resolução de equações diferenciais*
- 5- *Séries*
  - a. *Critérios de convergência*
  - b. *Séries trigonométricas e séries de potências*
- 6- *Aproximação de funções*
  - a. *Séries de Taylor*
  - b. *Séries de Fourier*

**4.4.5. Syllabus:**

- 1- *Differentiation*
  - a. *Applications to engineering problems*
  - b. *Limits*
- 2- *Integration*
  - a. *Indefinite integral*
  - b. *Definite integral*
  - c. *Fundamental Theorem*
  - d. *Integration Techniques*
  - e. *Application of integration - Areas and Volumes*
- 3- *Differential Equations*
  - a. *First Order Differential Equations*
  - b. *Second Order Differential Equations*
- 4- *Laplace Transform and their use to solve Differential Equations*
- 5- *Series*
  - a. *Convergence criteria*
  - b. *Trigonometric series, power series*
- 6- *Function approximation*
  - a. *Taylor series*
  - b. *Fourier series*

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

*O principal objetivo desta unidade curricular é fornecer aos estudantes conhecimentos em cálculo diferencial e integral a uma variável. Pretende-se que os estudantes tenham uma visão clara das principais áreas do cálculo, aprendam a resolver derivadas e integrais de funções, assim como usá-las em várias aplicações de engenharia, bem como aprender a usar diferentes técnicas de integração e equações diferenciais e usar séries e polinómios.*

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

*The main objective of this course unit was to provide the students with training knowledge of differential and integral calculus with one variable. It was intended that students have a clear view on key areas of this subject, such as solving derivatives of functions, integrals and use them in various engineering applications, use different integration techniques and differential equations and use series and polynomials.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*Nas aulas teóricas apresenta-se e discute-se o programa proposto a um nível teórico com o recurso a exemplos aplicados. As aulas teórico-práticas são destinadas à análise e resolução de problemas pelos estudantes. Os estudantes que já tenham obtido frequência na disciplina nos anos anteriores, estão dispensados da frequência das aulas podendo efetuar os minitests ou ir a recurso (de acordo com as normas gerais de avaliação em vigor na FEUP). Fórmula de avaliação: A classificação final considera a média das classificações nos dois minitests. Para obter aprovação é necessário uma média superior ou igual a 9.5 (em 20) e uma classificação mínima de 5 valores (em 20) em cada um dos minitests. O estudante que não tenha obtido aprovação, pode apresentar-se a Recurso para avaliação sobre a matéria de UM dos minitests à sua escolha ou sobre a totalidade da matéria. O estudante que já tenha obtido aprovação, pode apresentar-se a Recurso para avaliação sobre a TOTALIDADE da matéria.*

**4.4.7. Teaching methodologies (including students' assessment):**

*At the theoretical lectures, one presents and discusses the proposed program at a theoretical level with the support of applied examples. The theoretical and practical classes are intended for analysis and applied problem solved by the students. Type of evaluation: Distributed evaluation without final exam*

*Terms of frequency: In order to attend mini-tests or exam, students need to be in conformity with the general standards of evaluation of FEUP. Students repeating the course, do not need to attend classes.*

*Formula of evaluation: The grade will be calculated taking into account the average of two tests.*

*To obtain approval it is required an average higher than or equal to 9.5 (out of 20) and a minimum of 5 (out of 20) in each of the tests. The student that has not obtained approval can attend an “appeal exam” on the subject of ONE of the tests or on the entire matter. The student that has already obtained approval can attend the “appeal exam” assessing the TOTALITY of matter.*

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

*A unidade curricular funciona no regime de duas aulas teóricas com uma duração de 1h30 por semana, seguido de uma aula teórico-prática de 2h. Esta tipologia de aulas permite que o docente se dedique na parte inicial à exposição de matéria teórica, logo seguida de um exemplo prático de aplicação resolvido pelo docente. Subsequentemente, será proposto aos estudantes a resolução de problemas que abordem a temática subjacente ao conteúdo teórico ministrado. O atendimento aos estudantes por parte dos docentes existe ao longo de todo o semestre em horário pré-estabelecido. Nas épocas de testes os docentes disponibilizam nos dias antecedentes períodos de atendimento mais prolongados.*

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

*This unit presents two lectures of 1h30 per week, followed by a 2h lecture. This typology allows that teacher to begin with the theoretical exposition, followed by a practical example of application. Subsequently, students will solve problems that address the theme underlying by the theoretical content.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*António, C. (2017). Análise Matemática 1 - Conteúdo teórico e aplicações. AEFEUP. ISBN: 978-989-98632-3-1.*

*Madureira, M. L. (2018). Problemas de equações diferenciais ordinárias de Laplace. ISBN: 972-752-065-0.*

*Madureira, M. L. (2018). Problemas de integrais de linha e superfície e de séries de Fourier. Universidade do Porto. Faculdade de Engenharia. ISBN: 978-989-99559-2-9.*

*Apostol, T. M. (2007). Calculus, Volume I, One-Variable Calculus, with an Introduction to Linear Algebra (Vol. 1). John Wiley & Sons.*

*Banner, A. (2007). The calculus lifesaver: All the tools you need to excel at calculus. Princeton University Press.*

*Larson, R. E., Hostetler, R. P., & Edwards, B. H. (1994). Calculus with analytical geometry. DC Heath.*

### **Mapa IV - Análise Matemática II**

#### **4.4.1.1. Designação da unidade curricular:**

*Análise Matemática II*

#### **4.4.1.1. Title of curricular unit:**

*Mathematical Analysis II*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*M*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

*52 (26T + 26TP)*

#### **4.4.1.6. ECTS:**

*6*

#### **4.4.1.7. Observações:**

*<sem resposta>*

#### **4.4.1.7. Observations:**

*<no answer>*

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

*José Augusto Trigo Barbosa, T-39h (2 turmas); TP - 39 h (11 turmas)*

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

*Maria Luísa Romariz Madureira, T-13h (2 Turmas); TP-13h (11 Turmas)*  
*Alexandre Miguel Prior Afonso, TP-104h (11 Turmas)*  
*Ana Maria Azevedo Neves, TP-78h (11 turmas)*  
*Catarina Rosa Santos Ferreira de Castro, TP-52h (11 Turmas)*

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

*1- ENQUADRAMENTO (BACKGROUND) A unidade curricular tem um carácter didático/científico e introduz e desenvolve em termos teóricos diversos conceitos que serão ferramentas essenciais para apoio às restantes unidades curriculares.*  
*2- OBJETIVOS ESPECÍFICOS (SPECIFIC AIMS) Adquirir conhecimentos sobre cálculo diferencial e integral de funções reais e vetoriais de uma ou várias variáveis.*  
*3- CONHECIMENTO PRÉVIO (PREVIOUS KNOWLEDGE) Cálculo diferencial e integral, álgebra vetorial e geometria analítica*  
*4- DISTRIBUIÇÃO PERCENTUAL (PERCENTAGE DISTRIBUTION) Componente Científica: 75% Componente Tecnológica 25%*  
*5- RESULTADOS DA APRENDIZAGEM (LEARNING OUTCOMES) Parametrizar curvas e conhecer as suas propriedades; calcular integrais de linha ao longo dessas curvas. Discutir a continuidade de funções escalares a várias variáveis. Obter derivadas parciais e direcionais para campos escalares. Calcular derivadas de funções compostas. Calcular integrais duplos, triplos e de superfície.*

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

*1- BACKGROUND The main aim is to introduce fundamental mathematical concepts. These aspects form an educational background for other subjects in the curricula.*  
*2- SPECIFIC AIMS The students should acquire solid theoretical and practical training on the main concepts and results of differential and integral calculus of several variables, including the basic theorems of calculus.*  
*3- PREVIOUS KNOWLEDGE Functions and graphs. Differential and integral calculus. Vector algebra. Lines and planes.*  
*4- PERCENTAGE DISTRIBUTION Scientific component:75% Technological component:25%*  
*5- LEARNING OUTCOMES Partial and directional derivatives for real-valued and vector-valued functions. The chain rule for real-valued and vector-valued functions including implicit functions. Apply parametric curves and surfaces to calculate line and surface integrals.*  
*Students should deepen their knowledge on the concepts of double and triple integrals.*

**4.4.5. Conteúdos programáticos:**

*1- FUNÇÕES VETORIAIS. Propriedades. Curvas. Comprimento de arco. Curvatura. 2- FUNÇÕES A VÁRIAS VARIÁVEIS. Superfícies quádricas. Curvas de nível e superfícies de nível. Derivadas parciais. Limites e continuidade. 3- GRADIENTES. Diferenciabilidade e gradiente. Gradientes e derivadas direcionais. Teorema do valor médio. Regras de derivação em cadeia. Valores máximos e mínimos. Diferenciais. 4- INTEGRAIS DUPLOS E TRIPLOS. Integral duplo sobre uma região. Integrais duplos usando coordenadas polares. Integrais triplos. Coordenadas cilíndricas. Coordenadas esféricas. Jacobianos; mudança de variáveis na integração múltipla. 5- INTEGRAIS DE LINHA E INTEGRAIS DE SUPERFÍCIE. Integrais de linha. Propriedades. Integrais de linha em relação ao comprimento de arco. Teorema de Green. Parametrização de superfícies. Área de uma superfície. Integrais de superfície. Divergência e rotacional. Teorema da divergência. Teorema de Stokes.*

**4.4.5. Syllabus:**

*1- VECTOR FUNCTIONS. Properties. Curves. Arc length. Curvature. 2- FUNCTIONS OF SEVERAL VARIABLES. Quadric surfaces. Level curves and level surfaces. Partial derivatives. Limits and continuity. 3- GRADIENTS. Differentiability and gradient. Gradients and directional derivatives. The mean-value theorem. Chain rules. Maximum and minimum values. Differentials. 4- DOUBLE AND TRIPLE INTEGRALS. The double integral over a region. Evaluating double integrals using polar coordinates. Triple integrals. Cylindrical coordinates. Spherical coordinates. Jacobians; changing variables in multiple integration. 5- LINE INTEGRALS AND SURFACE INTEGRALS. Line integrals. Properties. Line integrals respect to arc length. Green's theorem. Parametrized surfaces. Surface area. Surface integrals. Divergence and curl. The divergence theorem. Stokes's theorem.*

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

*A unidade curricular tem dois objetivos fundamentais: por um lado, tratando-se de uma unidade curricular propedêutica tem um carácter didático/científico, promovendo o desenvolvimento do raciocínio lógico e de métodos de análise e, por outro, visa introduzir e desenvolver em termos teóricos um conjunto de conceitos que serão ferramentas essenciais para apoio às unidades curriculares mais específicas dos diferentes ramos da engenharia lecionados no DEIC. São introduzidos conceitos fundamentais sobre Análise Matemática, que são essenciais para a formação matemática de um estudante de Engenharia Informática e de Computadores.*

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

*The curricular unit has two fundamental objectives: on the one hand, in the case of a propaedeutic curricular unit, it has a didactic/scientific character, promoting the development of logical reasoning and analysis methods and, on the other hand, it aims to introduce and develop in theoretical terms a set of concepts that will be essential tools to support the more specific curricular units of the different branches of engineering taught at DEIC. Fundamental concepts about Mathematical Analysis are introduced, which are essential for the mathematical background of a student of Informatics and Computer Engineering.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*Nas aulas teóricas procede-se à exposição detalhada do programa da unidade curricular; sempre que possível são apresentados exemplos simples de aplicação. As aulas teórico-práticas são destinadas à resolução de problemas.*

*Tipo de Avaliação: Avaliação distribuída sem exame final*

*Condições de Frequência: Cumprir o disposto no art. 8º do READFEUP.*

*Fórmula de avaliação: Durante o processo de Avaliação Distribuída o estudante deve realizar duas provas escritas, cada uma delas com a duração de 2h. A classificação final é a média das classificações obtidas nas duas provas realizadas.*

*A obtenção de aprovação exige: cumprimento do disposto no art. 4º do READFEUP; classificação igual ou superior a 5,5 valores em qualquer uma das duas provas efetuadas.*

*No final do semestre o estudante pode ainda realizar uma prova de reavaliação para melhorar a classificação obtida no final do processo de Avaliação Distribuída.*

*A obtenção de uma classificação de 20 valores exige a realização de uma prova oral.*

**4.4.7. Teaching methodologies (including students' assessment):**

*Theoretical classes will be based on the presentation of the themes of the course unit. Theoretical-practical classes will be based on problem solving by students, where they have to apply tools and mathematical concepts taught in theoretical classes.*

*Type of evaluation: Distributed evaluation without final exam*

*Terms of frequency: Students cannot miss more classes than allowed in the regulation.*

*Formula of evaluation: The student must attend to two written exams, with the duration of 2 hours each. Final grade will be based on the average grade of the two exams.*

*In order to pass the course the student must: satisfy what is arranged in Art. 4 of General Evaluation Rules of FEUP; earn a grade of five point five out of twenty or better in each of the exams.*

*At the end of the semester students will be able to attend a new exam in order to improve their final grade.*

*The achievement of a classification of 20 requires an oral test.*

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

*A unidade curricular funciona no regime de uma aula teórico de 2 horas e uma aula teórico-prática de 2 horas, por semana. Na aula teórica o docente fará a exposição teórica dos conteúdos, complementando-a, sempre que possível, com a resolução de breves exemplos de aplicação. Na aula prática serão propostos exercícios sobre a matéria lecionada na aula teórica.*

*Durante o semestre cada docente disponibilizará um horário de atendimento semanal aos seus estudantes. Nas épocas de provas de avaliação os períodos de atendimento serão reforçados.*

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

*The curricular unit operates on a 2-hour theoretical class and a 2-hour theoretical-practical class per week. In the theoretical class the teacher will make the theoretical presentation of the contents, complementing it, whenever possible, with the resolution of brief application examples. In the practical class exercises will be proposed on the subject taught in the theoretical class.*

*During the semester each teacher will provide a weekly service schedule to their students. During periods of evaluation tests, the attendance periods will be reinforced.*

**4.4.9. Bibliografia de consulta/existência obrigatória:**

*Salas, S. L., Etgen, G. J., & Hille, E. (2006). Calculus: One and several variables. John Wiley & Sons.*

*Barbosa, J.A.T. (2018). Noções sobre Geometria Analítica e Análise Matemática. Efeitos Gráficos. ISBN: 978-989-99559-7-4.*

*Barbosa, J.A.T., (2020) Noções sobre Análise Matemática. Efeitos Gráficos. ISBN: 978-989-54350-4-3.*

*Barbosa, J.A.T., Apontamentos de apoio às aulas teóricas.*

*Madureira, M.L.R. (2017) Problemas de integrais de linha e superfície e de séries de Fourier. Efeitos Gráficos. ISBN: 978-989-99559-2-9.*

**Mapa IV - Arquitetura de Computadores****4.4.1.1. Designação da unidade curricular:**

*Arquitetura de Computadores*

**4.4.1.1. Title of curricular unit:**

*Computer Architecture*

**4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

**4.4.1.3. Duração:**

*Semestral*

**4.4.1.4. Horas de trabalho:**

162

**4.4.1.5. Horas de contacto:**

52 (26T + 26TP)

**4.4.1.6. ECTS:**

6

**4.4.1.7. Observações:**

&lt;sem resposta&gt;

**4.4.1.7. Observations:**

&lt;no answer&gt;

**4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Ricardo Jorge Gomes Lopes da Rocha (T - 26h - 2 turmas; TP - 52h - 12 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***João Paulo de Castro Canas Ferreira (T - 26h - 2 turmas; TP - 52h - 12 turmas)**António José Duarte Araújo (TP - 78h - 12 turmas)**Bruno Miguel Carvalhido Lima (TP - 52h - 12 turmas)**Gil Coutinho Costa Seixas Lopes (TP - 52h - 12 turmas)**José Maria Corte Real da Costa Pereira (TP - 26h - 12 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):**

1. *Explicar a interação do processador com os dispositivos de entrada / saída (E/S).*
2. *Explicar o uso de interrupções.*
3. *Avaliar as diferentes abordagens à gestão de E/S.*
4. *Descrever as características das atuais tecnologias de armazenamento secundário e avaliar o seu impacto sobre o desempenho.*
5. *Explicar as convenções de chamada de procedimento e escrever pequenos programas em C++/"assembly".*
6. *Descrever os recursos e as aplicações dos conjuntos de instruções SIMD.*
7. *Explicar "pipelining" básico no nível das instruções e o seu efeito no desempenho.*
8. *Explicar os mecanismos para lidar com dependências de dados e de controlo.*
9. *Descrever arquiteturas superescalares (emissão simultânea de instruções; execução fora de ordem).*
10. *Descrever organizações de memória "cache" e explicar o seu uso para melhorar o desempenho.*
11. *Discutir as limitações de desempenho dos processadores de núcleo único.*
12. *Descrever a organização básica de um processador de memória partilhada com vários núcleos.*

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

1. *Explain processor interaction with input/output (I/O) devices.*
2. *Explain the use of interrupts and discuss interrupt overhead and latency.*
3. *Evaluate the various approaches to I/O management (polling, interrupts, DMA).*
4. *Describe characteristics of current secondary storage technologies and their impact on performance.*
5. *Explain inter-language procedure-calling conventions and write small mixed C++/assembly programs.*
6. *Describe features and applications of SIMD instructions.*
7. *Describe cache memory organizations and explain the use of cache memory to improve performance.*
8. *Explain basic instruction-level pipelining, and the effect of pipelining on performance.*
9. *Explain how to mitigate the effect of pipeline hazards.*
10. *Describe superscalar architectures (multi-issue operation, and in-order and out-of-order execution).*
11. *Discuss the performance limitations of single-core processors.*
12. *Describe the basic organization of a multi/many-core, shared memory processor.*

**4.4.5. Conteúdos programáticos:**

- M1. *Introdução às plataformas computacionais (CPU, memória e periféricos).*
- M2. *Interface com periféricos: "polling", interrupções, DMA. Barramentos para comunicação com periféricos.*
- M3. *Subsistemas de armazenamento de dados (discos magnéticos, unidades de estado sólido, RAID).*
- M4. *Estimativa de desempenho de tarefas que envolvem atividade E/S significativa.*
- M5. *Convenções de chamada entre linguagens (invocação, passagem de parâmetro).*
- M6. *Instruções SIMD para paralelismo explícito de dados.*
- M7. *Organizações comuns de memórias "cache"; seu impacto no desempenho.*
- M8. *Paralelismo a nível de instruções: "pipelining"; limitações; gestão de dependências de dados e de controlo.*
- M9. *Paralelismo em nível de instruções: emissão de múltiplas instruções; execução fora de ordem.*
- M10. *Limitações de processadores de núcleo único; limitações de frequência e potência.*
- M11. *Organização básica de um processador multi-core; coerência e sincronização de cache.*

**4.4.5. Syllabus:**

- M1. Introduction to computational platforms (CPU, memory and peripherals).*
- M2. Peripheral interfacing: polling, interrupts, DMA. Buses for communication with peripherals.*
- M3. Data storage subsystems (magnetic disks, solid-state drives, RAID).*
- M4. Performance estimation of tasks involving significant I/O.*
- M5. Inter-language calling conventions (invocation, parameter passing).*
- M6. SIMD instructions for explicit data parallelism.*
- M7. Common cache memory organizations and their impact on performance.*
- M8. Instruction-level parallelism: pipelining; limitations; handling of data and control dependencies.*
- M9. Instruction-level parallelism: multiple instruction issue; out-of-order execution.*
- M10. Limitations of single-core processors; frequency and power walls.*
- M11. Basic organization of a multi-core processor; cache coherence and synchronization.*

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

*A correspondência entre conteúdos programáticos e os objetivos de aprendizagem é quase direta, conforme descrito a seguir.*

- M1: Objetivo 1.*
- M2: Objetivos 2 e 3.*
- M3: Objetivo 4.*
- M4: Objetivos 4.*
- M5: Objetivo 5.*
- M6: Objetivo 6.*
- M7: Objetivo 7.*
- M8: Objetivos 8 e 9.*
- M9: Objetivos 10.*
- M10: Objetivo 11.*
- M11: Objetivo 12.*

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

*The syllabus maps onto the learning objectives in a rather straightforward way, as described next:*

- M1: Objective 1.*
- M2: Objectives 2 e 3.*
- M3: Objective 4.*
- M4: Objectives 4.*
- M5: Objective 5.*
- M6: Objective 6.*
- M7: Objective 7.*
- M8: Objectives 8 e 9.*
- M9: Objective 10.*
- M10: Objective 11.*
- M11: Objective 12.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*A unidade curricular tem uma componente teórica baseada em aulas de exposição dos diversos temas acompanhados da apresentação de exemplos e respetiva discussão. As aulas teórico-práticas incluem a apresentação, análise e resolução de um conjunto de questões e de casos de estudo.*

*As atividades de aprendizagem a realizar fora do período de aulas são:*

- 1) Questionários de escolha múltipla (autoavaliação);*
- 2) Utilização de simuladores ou plataformas computacionais para avaliar o desempenho dos sistemas;*

*Tipo de Avaliação: Avaliação distribuída com exame final.*

*Fórmula de avaliação: A avaliação tem as seguintes componentes:*

- a) Um teste (T) com questões de escolha múltipla (1h30)*
- b) Dois trabalhos práticos de grupo (P1 e P2) com guião (2h00)*
- c) Exame final (E) com questões de resposta aberta (2h00)*

*A nota da avaliação distribuída é  $D = (P1+P2+T)/3$ .*

*A nota final é dada  $NF = D \times 0,6 + E \times 0,4$ .*

**4.4.7. Teaching methodologies (including students' assessment):**

*The course lectures on the various topics, which include the presentation and discussion of examples and discussion. The theoretical-practical sessions include the presentation, analysis and resolution of a number of problems and to scripted assignments.*

*Learning activities outside the classroom:*

- 1) Multiple-choice questionnaires for self-assessment;*
- 2) Use of simulators to evaluate system performance.*

*Assessment Type: Distributed assessment with final exam.*

*Evaluation Formula: The evaluation has the following components:*

- a) A test (T) with multiple choice questions (1h30)*
- b) Two group assignments (P1 and P2) with script (2h00)*

c) *Final exam (E) with open-ended questions (2h00)*

*Grade of the distributed evaluation is  $D = (P1 + P2 + T) / 3$ .*

*The course grade is given  $NF = D \times 0.6 + E \times 0.4$ .*

*The completion of this curricular unit requires that both following conditions be satisfied:*

- 1. Score D higher than 7.0 (out of 20);*
- 2. Score E higher than 7.0 (out of 20).*

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

*O teste avalia os conteúdos programáticos abordados nos módulos M1-M4 (essencialmente, o tratamento de periféricos). Os trabalhos com guião versam aspetos mais quantitativos da arquitetura de computadores (simulação de memórias cache e de funcionamento em pipeline ou avaliação de plataformas físicas) e permitem ao estudante adquirir alguma experiência com métodos quantitativos de avaliação de desempenho e adquirir intuição para o funcionamento dos subsistemas tratados. Também tornam a avaliação mais variada e distribuída ao longo do semestre.*

*O uso de simuladores e/ou ferramentas de avaliação de desempenho existentes nos sistemas Linux e/ou Windows 10 confere uma aplicabilidade imediata às competências adquiridas. Neste contexto, o uso ou desenvolvimento de “micro-benchmarks” em linguagem “assembly” constitui uma aplicação natural.*

*Os testes de autoavaliação voluntários permitem aos estudantes avaliar o estado dos seus conhecimentos e servem para despoletar dúvidas e revisões de assuntos específicos nas aulas teórico-práticas. Também constituem uma oportunidade para os estudantes refletirem sobre o processo de ensino/aprendizagem.*

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

*The test evaluates the syllabus covered in the modules M1-M4 (essentially the handling of peripherals). The practical tasks (done according to a script) cover more quantitative aspects of computer architecture (cache memory simulation and pipeline operation, or evaluation of actual computational platforms) and allow the student to gain some experience with quantitative methods of performance evaluation and to gain insight into the operation of the respective subsystems. They also make the assessment more diverse and better distributed along the semester.*

*The use of simulators and / or performance assessment tools existing on Linux and / or Windows 10 systems gives immediate applicability to the acquired skills. In this context, the use or development of micro-benchmarks in assembly language is a natural application.*

*Voluntary self-assessment tests allow students to assess the state of their knowledge and serve to trigger questions and revisions of specific subjects in theoretical-practical sessions. They also provide an opportunity for students to reflect on the teaching / learning process.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

- 1. Patterson, D., & Hennessy, J. (2016). Computer Organization and Design: The Hardware/Software Interface ARM Edition. Elsevier.*
- 2. Harris, S., & Harris, D. (2016). Digital Design and Computer Architecture ARM Edition. Elsevier.*
- 3. Jacob, B., Ng, S. W., & Wang, D. T. (2008). Memory Systems. Elsevier.*
- 4. Jain, A., & Lee, C. (2019). Cache Replacement Policies. Morgan & Claypool Publishers.*
- 5. Balasubramonian, R. (2019). Innovations in the Memory System. Morgan & Claypool Publishers.*

### **Mapa IV - Bases de Dados**

#### **4.4.1.1. Designação da unidade curricular:**

*Bases de Dados*

#### **4.4.1.1. Title of curricular unit:**

*Databases*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

*52 (26T + 26TP)*

#### **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):***Carla Alexandra Teixeira Lopes (T - 26h - 2 turmas; TP - 26h - 11 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Eduardo Resende Brandão Marques (T - 26h - 2 turmas; TP - 26h - 11 turmas)**João Miguel Rocha da Silva (TP - 78h - 11 turmas)**Michel Celestino Paiva Ferreira (TP - 52h - 11 turmas)**Docente(s) a contratar (TP - 104h - 11 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Após a conclusão desta unidade o estudante deve ser capaz de:*

- Criar um modelo conceptual para uma base de dados num contexto específico;
- Mapear um modelo conceptual para um esquema relacional;
- Identificar dependências funcionais numa relação;
- Diferenciar formas normais em esquemas relacionais;
- Decompor esquemas relacionais para a forma normal Boyce-Codd e 3a forma normal;
- Criar uma base de dados relacional usando a linguagem SQL a partir de um esquema relacional;
- Usar álgebra relacional para interrogar bases de dados;
- Consultar bases de dados relacionais usando a linguagem SQL;
- Compreender a necessidade de índices e identificar o melhor índice a criar numa dada situação;
- Usar transações e concorrência que garantam a integridade dos dados;
- Explicar as principais características de um sistema de gestão de bases de dados;
- Identificar as diferenças entre bases de dados relacionais, multidimensionais e NoSQL e qual a mais adequada a uma dada situação.

**4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***Upon completion of this unit students should be able to:*

- Define a high-level database model for a particular context;
- Map a high-level database model to a relational schema;
- Identify the functional dependencies of a relation;
- Differentiate normal forms in relational schemas;
- Decompose relational schemas to the Boyce-Codd normal form and the third normal form;
- Create a relational database in SQL from a relational schema;
- Use relational algebra to query databases;
- Query relational databases in SQL;
- Understand the need for indexes and identify the best index to create in a given situation;
- Use transactions and concurrency to ensure the data integrity of a database;
- Explain the main features of a database management system;
- Identify the differences between a relational database, a multidimensional database and a NoSQL database and be able to identify the most adequate for a particular situation.

**4.4.5. Conteúdos programáticos:***Introdução às bases de dados**Desenho de bases de dados**Modelação conceptual usando UML**Esquema relacional**Mapeamento do modelo conceptual para o esquema relacional**Dependências funcionais e normalização de esquemas relacionais**Criação de bases de dados**Linguagem de definição de dados SQL**Restrições de integridade**Interrogação e modificação de bases de dados**Álgebra relacional como linguagem de interrogação**Linguagem de manipulação de dados SQL**Criação e modificação de vistas**Gestão de bases de dados**Gatilhos**Índices**Transações, controlo de concorrência e recuperação**Controlo de acessos**Introdução a outros paradigmas de bases de dados**Armazéns de dados e OLAP**Sistemas NoSQL*

**4.4.5. Syllabus:**

*Introduction to Databases*  
*Database design*  
*Conceptual modeling using UML*  
*Relational schema*  
*Mapping the conceptual model to the relational schema*  
*Functional dependencies and normalization of relational schemas*  
*Database creation*  
*SQL Data Definition Language*  
*Constraints*  
*Database querying and modification*  
*Relational algebra as a query language*  
*SQL Data Manipulation Language*  
*Creation and modification of views*  
*Database management*  
*Triggers*  
*Indexes*  
*Transactions, concurrency, and recovery*  
*Authorization*  
*Introduction to other database paradigms*  
*Data warehousing and OLAP*  
*NoSQL systems*

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

*Os conteúdos programáticos estão alinhados com os objetivos de aprendizagem. Os primeiros 5 objetivos de aprendizagem são cobertos no tópico “Desenho de base de dados”. O 6º objetivo de aprendizagem é visto no tópico “Criação de base de dados”. Os objetivos 7 e 8 são abordados no tópico “Interrogação e modificação de bases de dados”. Os objetivos 9 e 10 estão associados ao tópico “Gestão de bases de dados”. O objetivo 11 é suposto ser atingido com a globalidade dos conteúdos programáticos. Por fim o último objetivo é visto no último tópico indicado nos conteúdos programáticos.*

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

*The syllabus is aligned with the learning objectives. The first 5 learning objectives are covered in the topic “Database design”. The 6th learning objective is seen in the topic “Database creation”. Objectives 7 and 8 are covered in the topic “Database querying and modification”. Objectives 9 and 10 are associated with the topic “Database management”. Goal 11 is supposed to be achieved with the overall syllabus. Finally, the last objective is seen in the last topic indicated in the syllabus.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*Utilizam-se as aulas teóricas para exposição dos assuntos do programa e exemplos de aplicação. As aulas práticas decorrem em laboratório, com recurso a sistemas de gestão de bases de dados. Nas aulas resolver-se-ão exercícios e efetuar-se-ão pequenos desenvolvimentos.*

*Para a obtenção de frequência, os estudantes não poderão exceder o número limite de faltas às aulas teórico-práticas. É necessário obter a classificação mínima de 8 valores (em 20) na componente de avaliação distribuída para acesso ao exame. A avaliação distribuída é determinada pela avaliação de um teste e um trabalho de grupo. O trabalho de grupo será avaliado em três momentos: as duas primeiras entregas valem 25% cada uma e a última entrega vale 50%.*

*Fórmula de avaliação:*

*T – Teste*

*PROJ - Projeto/Trabalho de grupo*

*EF - Exame Final (mínimo: 8 valores)*

*Nota = arredonda (0,2 T + 0,3 PROJ + 0,5 EF)*

**4.4.7. Teaching methodologies (including students' assessment):**

*Theoretical concepts will be transmitted to students in lectures. In lab classes, practical exercises will be performed, and the group work will be monitored. The project is an essential component of the evaluation because it is where students are expected to practice their skills more intensely. In the theoretical and lab classes, concrete cases of application of the theoretical concepts will be presented.*

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

*Os conceitos teóricos serão transmitidos aos estudantes nas aulas teóricas. Nas aulas práticas serão realizados exercícios de aplicação prática e será feito o acompanhamento ao trabalho de grupo, onde se espera que todas as competências sejam colocadas em prática de uma forma mais intensa. Nas aulas teóricas e nas aulas práticas serão apresentados casos concretos de aplicação dos conceitos teóricos.*

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

*Theoretical concepts will be transmitted to students in lectures. In lab classes, practical exercises will be performed, and the group work will be monitored. The project is an essential component of the evaluation because it is where*

*students are expected to practice their skills more intensely. In the theoretical and lab classes, concrete cases of application of the theoretical concepts will be presented.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Ullman, J. D., & Widom, J. (2008). A first course in database systems. Upper Saddle River (N.J.): Prentice Hall.*

*Ramakrishnan, R., & Gehrke, J. (2000). Database management systems. Boston, MA: McGraw-Hill.*

*Kimball, R., & Ross, M. (2009). The data warehouse toolkit the complete guide to dimensional modeling. New York, NY: Wiley.*

### **Mapa IV - Competências Transversais**

#### **4.4.1.1. Designação da unidade curricular:**

*Competências Transversais*

#### **4.4.1.1. Title of curricular unit:**

*Transferable Skills*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*DP/CTT/QACUP*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*40.5*

#### **4.4.1.5. Horas de contacto:**

*Para efeitos de cálculo, estimamos 13 horas de contacto*

#### **4.4.1.6. ECTS:**

*1.5*

#### **4.4.1.7. Observações:**

*Unidade Curricular a seleccionar entre as disponíveis no Catálogo de Competências Transversais/ Transferíveis da Universidade do Porto*

#### **4.4.1.7. Observations:**

*Course to be selected among those available in the Soft/Transferrable Skills Catalogue from the University of Porto.*

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

*Francisco Manuel Madureira e Castro Vasques de Carvalho (coordenador de CTs na FEUP)*

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

*O corpo docente será definido em função da CTE escolhida pelo estudante.*

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

- Reconhecer a importância de adotar uma atitude proactiva, comprometida e responsável;*
- Analisar as motivações e capacidades de trabalho em equipa;*
- Aplicar com sentido crítico diferentes atitudes comunicacionais em diferentes situações de interação;*
- Reconhecer a importância da ética no trabalho e do pensamento crítico, entre outras competências;*
- Produzir um relatório científico, de acordo com as normas da comunidade científica;*
- Saber adequar comportamentos a diferentes situações profissionais, pessoais e/ou relacionais;*
- Compreender a importância da cooperação e entreajuda para o clima organizacional;*
- Compreender os desafios inerentes à liderança de equipas, à tomada de decisão, à gestão de tempo e à colaboração com colegas de trabalho;*
- Compreender as mais-valias e desafios da criação do próprio emprego;*
- Identificar aspetos fundamentais da história, cultura e pensamento contemporâneo e da sua importância para o desenvolvimento de atitudes e comportamentos de cidadania.*

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

- To recognize the relevance of a proactive, committed and responsible attitude;*
- To analyse their own motivations and teamwork skills;*
- To critically apply different communication skills in different interaction situations;*

- To recognize the relevance of critical thinking and work ethics, among other skills;
- To produce adequate scientific reports, according to the scientific community standards;
- To adapt personal behaviours to different professional and/or relational situations;
- To understand the relevance of cooperation and mutual help for the organizational behaviour;
- To understand the major challenges related to team leadership, decision making, time management and collaboration with co-workers;
- To understand the challenges and advantages of their own job creation;
- To identify fundamental aspects of history, culture and contemporary thinking, and of their relevance for the development of citizenship behaviours.

#### 4.4.5. Conteúdos programáticos:

*Conteúdos programáticos específicos de cada Unidade Curricular disponível no Catálogo de Competências Transversais/ Transferíveis da Universidade do Porto.*

#### 4.4.5. Syllabus:

*Specific syllabus of each available course unit from the Soft/Transferrable Skills Catalogue of the University of Porto.*

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

*Específica de cada Unidade Curricular disponível no Catálogo de Competências Transversais/ Transferíveis da Universidade do Porto.*

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

*Specific for each available course unit, from the Soft/Transferrable Skills Catalogue of the University of Porto.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*Específica de cada Unidade Curricular disponível no Catálogo de Competências Transversais/ Transferíveis da Universidade do Porto.*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Specific for each available course unit, from the Soft/Transferrable Skills Catalogue of the University of Porto.*

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

*Perspetiva-se a oferta de diferentes tipos de CTs, organizadas de acordo com os seguintes tópicos:*

1. *Ações de formação em áreas transversais de Engenharia, incluindo:*
  - a. *Literacia científica, incluindo escrita de relatórios científicos com utilização de ferramentas bibliográficas;*
  - b. *Utilização de ferramentas transversais para análise de dados.*
2. *Ações de formação em soft skills relevantes para estudantes de Engenharia, incluindo:*
  - a. *Gestão de Tempo e Organização Pessoal;*
  - b. *Comunicação Assertiva;*
  - c. *Empregabilidade.*
3. *Ações de formação em Artes ou Ciências Sociais e Humanas relevantes para estudantes de Engenharia (oferecidas por outras Unidades Orgânicas da UPorto), incluindo:*
  - a. *Tópicos de Arte e Cultura Contemporânea (FBAUP);*
  - b. *Tópicos de Filosofia, Epistemologia, Ética aplicada à ciência e tecnologia, Pensamento crítico e Argumentação, Identidade e Interculturalidades (FLUP);*
  - c. *Tópicos de Psicologia do Trabalho e das Organizações, incluindo Gestão do Stress, Resolução de Problemas, Tomada de Decisão e Orientação para Resultados, Criatividade e Mudança Organizacional, Motivação de Equipas, Saúde e Bem-estar nas Organizações, Negociação e Gestão de Conflitos (FPCEUP).*
4. *Atividades estudantis com foco na formação em softskills (previamente certificadas pela UPorto), incluindo:*
  - a. *Organização das Jornadas de Engenharia Informática / Jornadas de Eletrotecnia / Jornadas de Engenharia Química / Jornadas de Engenharia Mecânica / etc. (creditável para membros da equipa organizadora, com limite ao n.º de organizadores);*
  - b. *Concurso "Pontes de Esparguete" / Concurso de Programação ACM / etc. (creditável para participantes e para membros da equipa organizadora, com limite ao n.º de organizadores);*
  - c. *Participação efetiva de estudantes em Organizações Internacionais de Estudantes de Engenharia.*
5. *Participação efetiva dos estudantes em Ciclos de Palestras, creditáveis como parte de Competências Transversais/ Transferíveis, de acordo com regras a especificar pela Universidade do Porto.*  
*Através da acima referida multiplicidade de oferta a disponibilizar pelo Catálogo de Competências Transversais/ Transferíveis da Universidade do Porto, será possível assegurar uma forte coerência entre as metodologias de ensino e os objetivos de aprendizagem da unidade curricular.*  
*Igualmente, será possível assegurar a cobertura de tópicos recomendados no "Computer Science Curricula 2013" do ACM e IEEE Computer Society, nomeadamente sobre comunicação e ética profissional. As escolhas dos estudantes serão sujeitas a validação pelo diretor do ciclo de estudos, podendo a inscrição numa das 3 unidades de Competências Transversais oferecidas (I, II e III) ser condicionada ao alinhamento com este referencial do ACM e IEEE.*

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

*We foresee the offer of multiple CT course units relevant to engineering students, organized according to the following topics:*

1. *Course units addressing transversal Engineering domains, including:*

- a. *Scientific literacy, including the writing of scientific reports, including the use of adequate bibliographic tools;*
- b. *Data analysis tools.*
- 2. *Soft skills course units, including:*
  - a. *Time Management and Personal Organization;*
  - b. *Assertive Communication;*
  - c. *Employability;*
- 3. *Course units addressing Art topics and/or Social and Human Science topics, which may be of interest for Engineering students (organized by the related UPorto Organic Units), including:*
  - a. *Topics of Contemporary Art and Culture (FBAUP);*
  - b. *Topics of Philosophy, Epistemology, Work Ethics applied to Science and Technology, Critical Thinking and Argumentation, Identity and Interculturalities (FLUP);*
  - c. *Topics in Work and Organizational Psychology, including Stress Management, Problem Solving, Decision Making, Creativity and Organizational Change, Team Motivation, Organizational Health and Well-Being, Negotiation and Conflict Management (FPCEUP);*
- 4. *Student activities focused on soft skills (previously certified by UPorto), including:*
  - a. *Organization of Informatics/Electrotechnical/Chemical/Mechanical Engineering Workshops (which may be creditable to organizing team members);*
  - b. *“Spaghetti Bridges” Contest / ACM Programming Contest / etc. (which may be creditable to both participants and organizing team members);*
  - c. *Student participation in International Engineering Student Organizations;*
- 5. *Student participation in Congresses, creditable as part of Transversal / Transferable Skills, according to a set of rules to be specified by the University of Porto.*  
*Considering the aforementioned CTs multiplicity available through the Catalogue of Transversal / Transferable Skills from the University of Porto, it is possible to ensure the required coherence level between teaching methodologies and learning outcomes of the curricular unit.*  
*Also, it will be possible to ensure coverage of topics recommended in the “Computer Science Curriculum 2013” of the ACM and IEEE Computer Society, namely communication and professional ethics. Student choices are subject to validation by the study cycle director, and the enrolment in one of the 3 units offered (I, II and III) conditioned upon alignment with this ACM and IEEE framework.*

#### 4.4.9. Bibliografia de consulta/existência obrigatória:

- Millar, D. C. (2011). Ready for take-off: a winning process for launching your engineering career. Upper Sadle River : Prentice Hall, cop. ISBN 978-0-13-608127-2*  
*Bibliografia complementar: Específica de cada Unidade Curricular disponível no Catálogo de Competências Transversais/ Transferíveis da Universidade do Porto.*

## Mapa IV - Compiladores

### 4.4.1.1. Designação da unidade curricular:

*Compiladores*

### 4.4.1.1. Title of curricular unit:

*Compilers*

### 4.4.1.2. Sigla da área científica em que se insere:

*EINFC*

### 4.4.1.3. Duração:

*Semestral*

### 4.4.1.4. Horas de trabalho:

*162*

### 4.4.1.5. Horas de contacto:

*52 (26T + 26TP)*

### 4.4.1.6. ECTS:

*6*

### 4.4.1.7. Observações:

*<sem resposta>*

### 4.4.1.7. Observations:

*<no answer>*

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

*João Manuel Paiva Cardoso (T - 26h - 2 turmas)*

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

*Rogério Ventura Lages dos Santos Reis (T - 26h - 2 turmas; TP - 52h - 11 turmas)*

*António Mário da Silva Marcos Florido (TP - 52h - 11 turmas)*

*Daniel Augusto Gama de Castro Silva (TP - 78h - 11 turmas)*

*Docente a contratar (TP - 104h - 11 turmas)*

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

*Fornecer os conceitos que permitam:*

- *compreender as fases de compilação de linguagens de programação, em especial de linguagens imperativas e de linguagens orientadas por objetos;*
- *especificar a sintaxe e semântica de linguagens de programação;*
- *compreender e utilizar as estruturas de dados e os algoritmos principais usados na implementação de compiladores.*

*As competências adquiridas permitirão aos estudantes:*

*LO1: desenvolver e implementar em software processadores de linguagens artificiais e de informação especificada textualmente segundo determinadas regras lexicais e sintáticas;*

*LO2 conceber e implementar em software as várias etapas relacionadas com compiladores, nomeadamente:*

- \* *expressões regulares e autómatos finitos;*
- \* *analísadores lexicais e sintáticos;*
- \* *analísadores semânticos;*
- \* *otimizações de código;*
- \* *geradores de código para microprocessadores ou para máquinas virtuais;*

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

*Provide concepts which allow to:*

- *understand the compilation phases of programming languages, in particular for imperative and object-oriented (OO) programming languages;*
- *specify the syntax and semantics of a programming language;*
- *understand and use the data structures and the main algorithms used to implement compilers.*

*The skills and learning outcomes will allow students to:*

*LO1: develop and implement in software, language processing systems of artificial languages and information textually specified under certain lexical and grammar rules;*

*LO2: design and implement in software the various compiler stages, namely:*

- \* *regular expressions and finite automata;*
- \* *lexical and syntactic analyzers;*
- \* *semantic analyzers;*
- \* *code optimization;*
- \* *code generation having processors or virtual machines as target;*

**4.4.5. Conteúdos programáticos:**

- *Introdução. Fases da compilação e estrutura típica de um compilador.*
- *Análise lexical. Expressões regulares e autómatos finitos.*
- *Análise sintática. Gramáticas. Implementação de analisadores sintáticos. Tratamento de erros.*
- *Análise semântica. Implementação de analisadores semânticos. Verificação de tipos.*
- *Ambientes de execução. Organização de memória e mecanismos para a passagem de parâmetros.*
- *Geração de código intermédio de baixo e de alto-nível.*
- *Seleção de instruções, escalonamento e alocação de registos.*
- *Técnicas de geração de código final e de otimização.*

**4.4.5. Syllabus:**

- *Introduction. Compilation phases and typical structure of a compiler.*
- *Lexical analysis. Regular expressions and finite automaton.*
- *Syntax analysis. Grammars. Syntax analysis' algorithms. Error handling.*
- *Semantic analysis. Type checking.*
- *Execution environments. Memory organization and schemes for parameter passing.*
- *High and Low-level intermediate representations. Intermediate code generation techniques.*
- *Instruction Selection, scheduling, and register allocation.*
- *Code optimizations and final code generation techniques.*

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

*As aulas de índole mais teórica apresentam os conceitos fundamentais de compiladores e soluções comuns para a sua conceção. Sendo complexos e variados, a sua sistematização é essencial para que os estudantes possam construir um modelo mental que lhes permita desenvolver um compilador. Sempre que possível, são usados exemplos ilustrativos na maioria dos casos oriundos de casos reais. No trabalho prático, os estudantes aplicam estes conceitos, reforçando deste modo a sua compreensão bem como das técnicas mais comuns associadas ao desenvolvimento de*

*um compilador. O trabalho a realizar foca normalmente o desenvolvimento de um mini-compilador, que integra a maioria das etapas de um compilador real.*

*A fórmula de cálculo da classificação final procura valorizar não só a capacidade de conceber/desenvolver um compilador, mas também a compreensão dos aspetos mais teóricos relacionados com compiladores.*

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

*The lectures are used to present the fundamental concepts of compilers and common solutions used in their design. Given the complexity and large body of knowledge, this material is presented in a well-organized and systematic fashion, to help the students to build a conceptual model that will help them to design a compiler. Whenever possible, illustrative examples are used, in most cases from real case studies.*

*In the programming assignment, the students have an opportunity to apply these concepts, reinforcing their understanding of these concepts and of the techniques most commonly used in the development of compilers. The work for the project is usually focused on the development of a mini-compiler, integrating the majority of the stages of a real compiler.*

*The formula used to compute the final grade reflects not only the ability to design/develop a compiler, but also the understanding of more theoretical aspects related to compilers.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*Aulas teóricas: exposição dos assuntos, acompanhada por exemplos e pela apresentação de pistas para o projeto do compilador. Aulas práticas: resolução de problemas e discussão de questões relacionadas com o trabalho prático.*

*Tipo de Avaliação: Avaliação distribuída sem exame final*

*AD: nota da avaliação distribuída (inclui trabalho prático) [0..20]*

*Condições de Frequência: Não ultrapassar limite máximo (3) de faltas nas aulas TP e AD >=10*

*Fórmula de avaliação:*

*Nota final (NF) = ROUND(0,60\*AD + 0,4\*T)*

*ÉPOCA NORMAL:*

*- T = T1 (nota no primeiro teste [0..20]) + T2 (nota no segundo teste [0..20])*

*O estudante obtém aproveitamento na UC na época normal se obtiver frequência, T1 >=7 e T2 >= 7 valores, MÉDIA(T1, T2) >= 8 valores e NF >= 10 valores.*

*ÉPOCA DE RECURSO:*

*- T = EX (nota no exame de recurso [0..20])*

*O estudante obtém aproveitamento na UC na época de recurso se obtiver frequência, EX >= 8 valores e NF >= 10 valores.*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Lectures: presentations, complemented by examples and clues for the lab work.*

*Labs: discussions and problem solving related to the practical work.*

*Type of evaluation: Distributed evaluation without final exam.*

*AD: grade obtained in the distributed evaluation (lab assignment) [0..20]*

*Terms of frequency: Finalizing the AD with a minimum grade of 10 (out of 20) and a maximum of three absences in the TP classes.*

*Formula of evaluation:*

*Final Grade (FG) = ROUND(0,60\*AD + 0,4\*T)*

*FIRST ROUND ("Época Normal"):*

*- T = T1 + T2 (grades obtained in the first and second test [0..20])*

*Each student will pass in the course in the "época normal" if he/she is according to the terms of frequency, obtained T1 >= 7 and T2 >= 7, an AVG(T1, T2) >= 8, and FG >= 10.*

*SECOND ROUND ("Época de recurso"):*

*- T = EX (grade obtained in the exam [0..20])*

*Each student will pass in the course in the "época de recurso" if he/she is according to the terms of frequency, obtained EX >= 8, and FG >= 10.*

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

*Esta unidade curricular apresenta uma simbiose muito forte entre a componente teórica e a componente prática. As aulas teóricas de exposição são utilizadas para focar os tópicos curriculares relacionados com compiladores e para que os estudantes conheçam muitas das técnicas utilizadas em compiladores.*

*É muito importante que os estudantes possam implementar algumas dessas técnicas e possam tomar conhecimento do processo de desenvolvimento de compiladores. Nesse sentido, uma componente relevante do ensino é baseada no desenvolvimento de um compilador, o que motiva os estudantes a estudar os problemas com que se confrontam, a procurar soluções e a aplicá-las ao seu caso concreto. Os estudantes, distribuídos em grupos, desenvolvem um mini-compilador completo (tipicamente um compilador para um subconjunto de uma linguagem de programação conhecida) que lhes possibilita adquirir e aplicar uma parte substancial dos conhecimentos teóricos adquiridos durante o semestre.*

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

*This curricular unit presents a strong symbiosis between the theoretical and the practical components.*

*The lectures are focused on topics related to compilers with the main objective to allow students to acquire skills related to techniques used in compilers.*

*It is very important that students implement some of those techniques and face the process of developing a compiler. A relevant component of the teaching is based on the development of a compiler. This motivates students to study the problems they face during the development, to search for solutions, and to apply those solutions to their specific cases. Students, organized in groups, develop a complete mini-compiler (usually considering a subset of a well-known programming language), which allows them to acquire and to apply a substantial part of the topics focused during the unit.*

**4.4.9. Bibliografia de consulta/existência obrigatória:**

*Appel, A. W. (2002). Modern Compiler Implementation in Java (2nd Edition). Cambridge University Press. ISBN: ISBN 0-521-82060-X.*

*Aho, A., Lam, M., Sethi, R., Ullman, J. (2007). Compilers: Principles, Techniques, and Tools (2nd Edition). Addison Wesley. ISBN: 0321486811.*

*Cooper, K. D., Torczon, L. (2011). Engineering a compiler (2nd Edition). Morgan Kaufmann. ISBN: 9780120884780, 9780080916613.*

*Louden, K. C. (1997). Compiler construction. Cengage Learning. ISBN: 0-534-93972-4*

*Muchnick, S. (1997). Advanced Compiler Design and Implementation. Morgan Kaufman Publishers. ISBN: ISBN 1-55860-320-4.*

**Mapa IV - Computação Gráfica****4.4.1.1. Designação da unidade curricular:**

*Computação Gráfica*

**4.4.1.1. Title of curricular unit:**

*Computer Graphics*

**4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

**4.4.1.3. Duração:**

*Semestral*

**4.4.1.4. Horas de trabalho:**

*162*

**4.4.1.5. Horas de contacto:**

*52 (26T + 26TP)*

**4.4.1.6. ECTS:**

*6*

**4.4.1.7. Observações:**

*<sem resposta>*

**4.4.1.7. Observations:**

*<no answer>*

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

*António Augusto de Sousa (T - 26h - 2 turmas)*

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

*Verónica Costa Teixeira Pinto Orvalho (T - 26h - 2 turmas; TP - 52h - 10 turmas)*

*António Fernando Vasconcelos Cunha Castro Coelho (TP - 78h - 10 turmas)*

*Rui Pedro Amaral Rodrigues (TP - 52h - 10 turmas)*

*Teresa Carla de Canha e Matos (TP - 78h - 10 turmas)*

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

*ENQUADRAMENTO: Nesta unidade curricular, efetua-se uma abordagem às matérias de Computação Gráfica segundo uma filosofia tipo Top-Down, iniciando-se com os temas de 3D (síntese de imagem, modelação), programação de*

*GPU's e terminando com a visita a vários algoritmos no âmbito dos 2D. A componente 3D da matéria é acompanhada, nas aulas práticas, com exercícios baseados em tecnologias de grande utilização, como sejam o OpenGL e o WebGL.*

**OBJETIVOS ESPECÍFICOS:**

- Transmitir conceitos, técnicas, algoritmos, tecnologias e arquiteturas de Computação Gráfica.
- Reforçar os conhecimentos teóricos com a sua aplicação prática, por meio da implementação, teste e avaliação de algoritmos abordados em teoria.

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

*BACKGROUND: In this course, the approach to computer graphics is made under a Top-Down philosophy, starting with the subjects most related to 3D (image synthesis, modelling), through GPU programming, and ending with a visit to several most basic algorithms in 2D. The 3D components of the programme are accompanied, in practical lessons, with exercises based on the usual technologies, like OpenGL and WebGL.*

**SPECIFIC AIMS**

- Transmit knowledge of concepts, techniques, algorithms, computer graphics technologies and architectures.
- Strengthen the theoretical knowledge with practical application, through the implementation, testing and evaluation of algorithms discussed in theory.

**4.4.5. Conteúdos programáticos:**

- 1) Panorâmica da Computação Gráfica e suas aplicações.
- 2) Síntese de Imagem de cenas 3D:
  - a) iluminação local e iluminação global;
  - b) iluminação suavizada, texturas;
  - c) cálculo de visibilidade;
  - d) projeção de sombras.
- 3) Representação de cor em Computação Gráfica
  - a) percepção da cor;
  - b) modelos de representação da cor.
- 4) Transformações geométricas 2D/3D.
- 5) Modelação
  - a) malhas poligonais 3D;
  - b) curvas;
  - c) superfícies e sólidos.
- 6) Linguagens de programação de "Shaders" em GPU's
- 7) Interação; conceitos breves para o desenvolvimento de interfaces gráficas.
- 8) Computação Gráfica 2D
  - a) algoritmos de rasterização de linhas;
  - b) algoritmos de rasterização de regiões.

**4.4.5. Syllabus:**

- 1) Computer Graphics panoramic and applications.
- 2) 3D image synthesis:
  - a) local lighting and global lighting;
  - b) smooth shading, textures;
  - c) visibility calculation;
  - d) shadows projection.
- 3) Colour representation in Computer Graphics
  - a) colour perception;
  - b) models for colour representation.
- 4) Geometric transformations: 3D/ 2D.
- 5) Modelling
  - a) 3D meshes;
  - b) curves;
  - c) surfaces and solids.
- 6) Programming languages of "Shaders" in GPU's
- 7) Interaction; concepts for the development of graphical interfaces.
- 8) 2D computer graphics
  - a) rasterization of lines;
  - b) rasterization of regions

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

*Os conteúdos programáticos focam os temas principais inerentes ao objetivo de desenvolver competências práticas, no âmbito da Computação Gráfica. Os cinco primeiros capítulos são orientados para o cumprimento dos objetivos relacionados com o aprofundamento teórico de assuntos relacionados com Computação Gráfica 3D em geral. O capítulo 6 destina-se a cobrir temas relacionados com placas gráficas programáveis, num contexto das atuais tecnologias disponíveis. Os capítulos seguintes dão cumprimento aos objetivos mais relacionados com interação, assim como uma breve visita a alguns algoritmos 2D. Em geral, o programa é estabelecido de forma a alargar os horizontes dos estudantes, aprofundando conhecimentos*

de outras áreas científicas, destinados a dar apoio aos aspetos multidisciplinares enunciados nos objetivos específicos.

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

*The syllabus focus on the main subjects inherent to the objective of developing practical skills, within the scope of Computer Graphics. The first five chapters are oriented to the fulfilment of the objectives related to the deepening of theoretical subjects related to 3D Computer Graphics in general. Chapter 6 cover the subjects related with programmable graphics boards, in the context of the available technologies. The following chapters give fulfillment to the objectives more related to interaction, as well as to some important 2D algorithms.*

*In general, the program is established in order to broaden the students horizons, deepening their knowledge of other scientific areas, aimed at supporting the multidisciplinary aspects set out in the specific objectives.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*Aulas teóricas:*

- Apresentação das principais teorias, técnicas e algoritmos utilizados em Computação Gráfica.

*Aulas Teórico-Práticas:*

- Discussão e desenvolvimento de exercícios, testados em computador, ao nível de cada grupo, com possíveis desenvolvimentos extra-escolares.

- Utilização de OpenGL/WebGL para verificação de alguns tópicos abordados teoricamente.

- Realização de dois projetos agregadores.

*Tipo de Avaliação: Avaliação distribuída com exame final*

*Condições de Frequência: Não exceder o número limite de faltas e obter um mínimo de 40% na classificação da avaliação distribuída.*

*Fórmula de avaliação: Legenda:*

*CF: Classificação Final*

*AvEx: Classificação de Exame*

*AvDis: Avaliação Distribuída*

*CF = 50% AvEx + 50% AvDis*

*AvDis = 40% Minitestes + 60% Projetos*

*Para obter aprovação é exigido um mínimo de 40% em qualquer das duas componentes de avaliação, distribuída e exame final.*

*NOTA: os mini-testes são sem consulta; o exame final é com consulta.*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Theoretical classes:*

- Presentation of the main theories, techniques and algorithms, used in computer graphics

*Theoretical-practical classes:*

- Exercises solving and discussion, tested in computer by each group with possible extra-scholar developments.

- Use of OpenGL/WebGL to verify some of the 3D theoretical topics.

- Development of two aggregating project.

*Type of evaluation: Distributed evaluation with final exam*

*Terms of frequency: Not exceed the absence limit and obtain a minimum of 40% in the distributed evaluation classification.*

*Formula of evaluation: Legend:*

*FG: Final Grade*

*ExEv: Exam grade*

*DisEv: Distributed Evaluation*

*FG = 50% ExEv + 50% DisEv*

*DisEv = 40% Mini-tests + 60% Projects*

*To pass, the student must have a minimum of 40% in any of the two evaluation components: distributed evaluation and final exam.*

*NOTE: Minitests are closed book; final exam is open book.*

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

*Todos os capítulos previstos no programa da UC contêm informação acerca dos principais conceitos, técnicas, algoritmos, tecnologias e arquiteturas de Computação Gráfica. As matérias abordadas nas aulas práticas cobrem a componente de implementação, teste e avaliação de algoritmos, reforçando os conhecimentos teóricos adquiridos.*

*No final da unidade curricular, após terem desenvolvido aplicações gráficas com base em tecnologias adequadas, os estudantes tomam conhecimento com os principais módulos de um sistema gráfico 3D, com a programação por eventos, assim como desenham e implementam interfaces gráficas, nas suas várias vertentes.*

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

*All the chapters included in the course program contain information about the main concepts, techniques, algorithms, technologies and architectures of Computer Graphics. The subjects covered in the practical classes cover the component of implementation, testing and evaluation of algorithms, reinforcing the theoretical knowledge acquired.*

*At the end of the course, after having developed graphical applications based on appropriate technologies, the students learn about main modules of a 3D graphic system, events programming, as well as designing and implementing graphical interfaces, in their various aspects.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Foley, J. D., van Dam, A., Feiner, S. & Hughes, J. (1994). Computer Graphics: Principles and Practice. Reading, MA: Addison-Wesley. ISBN: 978-0-201-12110-0.*

*Pereira, J., Brisson, J., Coelho AS., Ferreira A. & Gomes M. (2018). Introdução à Computação Gráfica. FCA. ISBN: 978-972-722-877-5.*

*Baker M. & Hearn D. (1996). Computer Graphics, C Version: International Edition. Prentice Hall. ISBN: 0-13-578634-7.*

### **Mapa IV - Computação Paralela e Distribuída**

#### **4.4.1.1. Designação da unidade curricular:**

*Computação Paralela e Distribuída*

#### **4.4.1.1. Title of curricular unit:**

*Parallel and Distributed Computing*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

*52 (26T + 26TP)*

#### **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):**

*Rolando da Silva Martins (T - 26h - 2 turmas; TP - 26h - 11 turmas)*

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

*Jorge Manuel Gomes Barbosa (T - 26h - 2 turmas; TP - 78h - 11 turmas)*

*Inês Castro Dutra (TP - 52h - 11 turmas)*

*Pedro Alexandre Guimarães Lobo Ferreira do Souto (TP - 26h - 11 turmas)*

*Docente a contratar (TP - 104h - 11 turmas)*

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

*- Distinguir o uso de recursos computacionais para obter uma resposta mais rápida pelo acesso eficiente a um recurso partilhado.*

*- Aplicar a sincronização apropriada num programa paralelo específico.*

- Identificar oportunidades para particionar um programa serie em módulos paralelos independentes.
- Definir "speedup" e explicar a noção de escalabilidade de um algoritmo a esse respeito.
- Explicar as implicações da latência de comunicação, falhas parciais e assincronia numa aplicação distribuída.
- Implementar uma aplicação distribuída baseada em threads e orientada a eventos.
- Identificar os riscos de segurança em aplicações distribuídas que podem ser mitigados por canais seguros e explicar os protocolos criptográficos usados.
- Identificar aplicações práticas de algoritmos básicos de sincronização, como eleições, e explicar porque pequenas alterações nesses algoritmos, podem quebrá-los.
- Avaliar os compromissos entre desempenho, tolerância a falhas e consistência em serviços replicados.

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

- Distinguish using computational resources for a faster answer from managing efficient access to a shared resource.
- Apply appropriate synchronization in a specific parallel program.
- Identify opportunities to partition a serial program into independent parallel modules.
- Define "speed-up" and explain the notion of an algorithm's scalability in this regard.
- Explain the implications of communication latency, partial-failures and asynchrony in a distributed application.
- Implement a distributed application using both thread-based and event-driven concurrency.
- Explain the main security risks in a distributed application, identify which can be mitigated by secure channels and which security protocols and cryptographic primitives these use.
- Identify problems in which basic synchronization algorithms such as elections can be used, and explain why small changes may break them.
- Assess the trade-offs among performance, fault-tolerance and consistency in replicated services.

#### 4.4.5. Conteúdos programáticos:

*Introdução à computação paralela. Medidas de desempenho. Máquinas paralelas. Organização de memória e efeito da gestão da memória cache no desempenho do processador. Limitações da computação paralela (Amdahl Law). Tipos de paralelismos: funcional, de dados e em "streams". Metodologias para paralelização de algoritmos. Modelos de programação paralela: Memória Partilhada e Memória Distribuída. Problemas de concorrência. Programação de multi-cores pelo modelo de memória partilhada utilizando OpenMP. Modelos de Computação. Introdução à computação distribuída. Modelos de comunicação: troca de mensagens, invocação remota de funções. Comunicação multicast. Processamento em sistemas distribuídos. Serviços com manutenção de estado. Implementações concorrentes baseadas em eventos vs. "threads". Segurança. Ameaças e ataques. Primitivas criptográficas. Canais de Comunicação seguros. Tolerância a falhas. Algoritmos de eleição. Protocolo "two-phase commit" Replicação e modelos de consistência.*

#### 4.4.5. Syllabus:

*Introduction to parallel computing.. Performance measures. Parallel architectures. Memory organization and the effect of cache management on processor performance. Amdahl Law. Ways of extracting parallelism: functional, data parallelism and streaming. Methodologies for developing parallel programs. Parallel programming models: shared memory and distributed model. Concurrency. Multi-core programming with OpenMP. Computational models and performance measures.*

*Introduction to distributed systems and network computing. Communication paradigms: message passing, remote procedure call and multicasting. Security in distributed systems: threats, cryptographic primitives and secure channels. Processing in distributed systems: state-full vs. stateless protocols, event-based and thread-based concurrency. Fault-tolerance: algorithms for leader election and two-phase commit. Replication and consistency.*

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

*A apresentação dos objetivos do paralelismo, a necessidade de usar várias unidades de computação (núcleos) e as metodologias para resolver um problema específico para obter uma versão paralela, leva à compreensão de como obter respostas mais rápidas, resolver problemas de sincronização e partição de problemas. As medidas de desempenho abordam o problema da análise de speedup, eficiência e escalabilidade. Para cada tópico do programa relacionado com computação distribuída (os últimos seis), apresentamos um objetivo de aprendizagem de alto nível. Para facilitar a correspondência, os objetivos são apresentados pela mesma ordem que o tópico correspondente. Por exemplo, para o tópico segurança, declaramos um resultado correspondente de aprendizagem de alto nível. A única exceção a esta regra diz respeito aos tópicos sobre comunicação e processamento em sistemas distribuídos, cujos resultados de aprendizagem são sintetizados na capacidade de implementar aplicações distribuídas concorrentes.*

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

*Introducing the goal of parallelism, the need for using several computing units (cores) and methodologies to address a particular problem to obtain a parallel version, lead to the understanding of achieving faster answers, synchronization problems and problem partition. The performance measures address the problem of speedup, efficiency and scalability analysis. For almost each syllabus' topic related to distributed computing (the last six), we present one high-level learning outcome. These are presented in the same order to make it easier to match the learning outcome to the topic. For*

*example, for the security topic, we state a matching high-level learning outcome. The only exception to this rule concerns the topics on communication and processing in distributed systems, whose learning outcomes are synthesized in the ability to implement concurrent distributed applications.*

#### **4.4.7. Metodologias de ensino (avaliação incluída):**

*As aulas teóricas são de exposição. Sempre que necessário, exemplos que ajudem à compreensão dos tópicos serão apresentados. A matéria exposta segue de perto a bibliografia principal recomendada. Para complementar, em tópicos muito pontuais, serão fornecidos apontamentos ou artigos.*

*Ao longo do semestre serão propostos problemas de programação e outros para consolidação dos conceitos apresentados nas aulas teóricas. Estes problemas serão discutidos nas aulas teórico-práticas, mas espera-se que os estudantes os resolvam fora das aulas. Os estudantes deverão ainda realizar 2 pequenos projetos, onde deverão aplicar os conceitos apresentados nas aulas teóricas.*

*Fórmula de avaliação: 0,45 PP + 0,1PA + 0,45ET*

*onde:*

*PP - Nota global dos projetos*

*PA - Classificação da participação nas Aulas/Discussões*

*ET - Classificação do Exame Teórico*

*Para aprovação à disciplina, os estudantes deverão ter uma classificação mínima de 40% quer no exame teórico quer no projeto.*

#### **4.4.7. Teaching methodologies (including students' assessment):**

*All topics are covered in the lectures, which are mostly expository and usually use transparencies. To motivate the students, we use real-world examples. The course contents follows closely the mandatory bibliography. Whenever necessary, class notes and/or articles will be provided.*

*The students have to solve both paper-and-pencil problems and small programming problems, which can be completed in a few hours. In addition, they have to do two small projects. Help for solving these problems and the project is given in the lab sessions.*

*Student evaluation is based on:*

*1) the projects (Pr)*

*2) the final exam (FE)*

*3) participation in class (PC)*

*The final grade formula is:*

*0.45 Pr + 0.45 FE + 0.1 PC*

*To successfully complete this course, students must have a minimum score of 40% in each of the Pr and FE components.*

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

*Todos os tópicos da unidade curricular, os quais são coerentes com os objetivos de aprendizagem, como argumentamos acima, são apresentados nas aulas teóricas. O grau de profundidade depende do nível de domínio esperado do objetivo correspondente.*

*Tópicos fundamentais de computação paralela tal como o desenvolvimento de programas de memória partilhada e avaliação do desempenho são praticados e consolidados resolvendo problemas de programação específicos.*

*Tópicos fundamentais de computação distribuída tal como o impacto da latência na comunicação, avarias-parciais e assincronia são apresentados repetidamente em diferentes tópicos, começando com uma panorâmica geral na primeira aula e depois com exemplos mais concretos à medida que aparecem no âmbito da discussão de outros tópicos.*

*Para objetivos em que se exige um maior domínio, propomos exercícios de papel-e-lápis. que exigem que os estudantes aprofundem o tópico (seja um algoritmo, protocolo ou técnica), ou problemas de programação, p.ex. para comunicação multicast ou o estabelecimento de canais seguros.*

*Naturalmente para objetivos que requerem que os estudantes saibam fazer, propomos também a realização de problemas de programação.*

*O projeto de desenvolvimento duma aplicação distribuída está diretamente ligado ao objetivo:*

*- "Implementar uma aplicação distribuída baseada em threads e orientada a eventos."*

*Através da integração de outros tópicos, tais como segurança ou replicação, este projeto também contribui para os objetivos de aprendizagem correspondentes.*

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

*In the lectures, we present all the topics of the course, which as argued above are coherent with the expected learning outcomes. The presentation depth of each topic depends on the expected mastery level of the respective learning outcome.*

*Key aspects of parallel computing such as developing efficient shared memory programs and performance evaluation in terms of speedup and efficiency, are consolidated and trained by developing specific programming exercises.*

*Key aspects of distributed computing such as the impact of communication latency, partial-failures and asynchrony on distributed applications are presented repeatedly in different topics, starting with a general overview in the first lecture on distributed computing, and then with more concrete examples as they arise in the discussion of the different topics.*

*Furthermore, for learning outcomes whose mastery level is higher, we assign either paper-and-pencil problems, that require the students to better understand the topic (be it an algorithm, a protocol, or a technique), or programming problems, e.g. multicast or secure channels.*

*Programming assignments are also used for learning outcomes that require knowing how to do, such as implementing a program or applying a parallel decomposition technique.*

*The project of development of a distributed application is directly related to the following learning outcome: "Implement a distributed application using both thread-based and event-driven concurrency." By integrating other topics, e.g. security or replication, this project also contributes to the respective learning outcomes.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*van der Pas, R., & Stotzer, E., & Terboven, C., (2017). Using OpenMP—The Next Step: Affinity, Accelerators, Tasking, and SIMD. The MIT Press.*  
*Quinn, M. J. (2017). Parallel programming in C with MPI and openMP. McGraw-Hill.*  
*van Steen, M., & Tanenbaum, A. (2017). Distributed systems (3rd ed.). Maarten van Steen.*

### **Mapa IV - Desenho de Algoritmos**

#### **4.4.1.1. Designação da unidade curricular:**

*Desenho de Algoritmos*

#### **4.4.1.1. Title of curricular unit:**

*Algorithm Design*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

*52 (26T + 26TP)*

#### **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):**

*Rosaldo José Fernandes Rossetti (T - 26h - 2 turmas; TP - 78h - 11 turmas)*

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

*Pedro Manuel Pinto Ribeiro (T - 26h - 2 turmas; TP - 52h - 11 turmas)*  
*Ana Paula Nunes Gomes Tomás (TP - 52h - 11 turmas)*  
*Docente a contratar (TP - 104h - 11 turmas)*

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

*Esta unidade curricular complementa e aprofunda os conhecimentos assimilados na unidade curricular de Algoritmos e Estruturas de Dados, pela introdução de técnicas de conceção de algoritmos para a resolução de diferentes tipos de problemas.*

*No final da unidade curricular, espera-se que o estudante seja capaz de:*

- a) conhecer e saber aplicar técnicas genéricas de conceção de algoritmos;*
- b) conhecer e saber aplicar algoritmos avançados em grafos;*
- c) conhecer e saber aplicar algoritmos em cadeias de caracteres;*

- d) identificar problemas intratáveis e algoritmos que fornecem soluções aproximadas;
- e) identificar problemas de otimização matemática e conhecer e saber aplicar algoritmos para resolução de alguns tipos de problemas de otimização linear e não linear.

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

*This course deepens the knowledge acquired in the previous curricular unit of Algorithms and Data Structures, by introducing techniques for devising and implementing algorithms to solve different classes of problems.*

*At the end of the course, the student is expected to be able to::*

- a) be acquainted with and know how to apply generic algorithm design techniques;
- b) be acquainted with and know how to apply advanced algorithms in graphs;
- c) be acquainted with and know how to apply algorithms in strings;
- d) be able to identify intractable problems and to devise algorithms to produce approximate solutions;
- e) be able to identify mathematical optimization problems and know and be able to apply algorithms for solving some types of linear and non-linear optimization problems.

#### 4.4.5. Conteúdos programáticos:

1. Técnicas de conceção de algoritmos: divisão e conquista; algoritmos gananciosos; programação dinâmica; algoritmos de retrocesso; aplicações.
2. Algoritmos avançados em grafos: fluxo máximo e fluxo de custo mínimo em redes de transporte; circuito de Euler e problema do carteiro chinês; problemas de emparelhamento e casamentos estáveis.
3. Algoritmos em strings: pesquisa exata; pesquisa aproximada; compressão de texto.
4. Problemas intratáveis: teoria dos problemas NP-completos; redução de problemas; exemplos de problemas intratáveis; algoritmos polinomiais aproximados.
5. Programação linear: problemas de otimização com restrições; resolução pelo algoritmo simplex.
6. Algoritmos numéricos: determinação de raízes de funções; resolução de sistemas de equações lineares e não lineares; determinação de mínimos locais pelo método da descida de gradiente; análise do erro; aplicações.

#### 4.4.5. Syllabus:

1. Algorithm design techniques: divide and conquer; greedy algorithms; dynamic programming; backtracking algorithms; application examples.
2. Advanced graph algorithms: maximum flow and minimum cost flow in transport networks; Euler circuit and the Chinese postman problem; matching and stable marriage problems.
3. String algorithms: exact string matching; approximate string matching; text/file compression.
4. Intractable problems: NP-complete problems theory; problem reduction techniques; examples of intractable problems; approximate polynomial algorithms.
5. Linear programming: constrained optimization problems; solution by the simplex algorithm.
6. Numerical algorithms: root finding; solving systems of linear and non-linear equations; finding local minima by the gradient descent method; error analysis; applications.

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

*Os conteúdos estão mapeados diretamente com os objetivos da seguinte forma:*

- 1 -> a
- 2 -> b
- 3 -> c
- 4 -> d
- 5 -> e (para problemas de otimização linear com restrições)
- 6 -> e (para problemas de otimização não-linear sem restrições)

*As técnicas genéricas de conceção de algoritmos são estudadas no 1º módulo, habilitando os estudantes a conceber novos algoritmos com base nessas técnicas. De seguida são estudados algoritmos eficientes em grafos e em strings, habilitando os estudantes a resolver eficientemente problemas práticos por aplicação de técnicas e algoritmos conhecidos. De seguida, são abordados problemas intratáveis conhecidos, técnicas de análise de problemas e alguns algoritmos aproximados, habilitando os estudantes a enfrentar problemas mais complexos. Finalmente, são estudadas técnicas específicas (oriundas da Investigação Operacional e da Análise Numérica) para resolução de problemas formuláveis como problemas de otimização matemática, linear e não linear, com ou sem restrições.*

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

*The contents are mapped directly with the objectives as follows:*

- 1 -> a
- 2 -> b
- 3 -> c
- 4 -> d
- 5 -> e (for constrained linear optimization problems)
- 6 -> e (for non-constrained, non-linear, optimization problems)

*Generic algorithm design techniques are studied in the 1st module, enabling students to design new algorithms based on these techniques. Then, efficient algorithms in graphs and strings are studied, enabling students to efficiently solve practical problems, adapting known techniques and algorithms. Then, intractable known problems, problem analysis techniques and some approximate algorithms are addressed, enabling students to face more complex problems. Finally, it are studied specific technics (from Operations Research and Numerical Analysis) for solving problems that can be formulated as mathematical optimization problems, linear and non-linear, constrained or non-constrained.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*As aulas teóricas são usadas para a exposição formal da matéria, acompanhada da apresentação de exemplos e sua discussão. As aulas práticas são usadas para a resolução de exercícios e desenvolvimento de pequenos programas (em C++, Java ou Python) para testar os algoritmos desenvolvidos.*

*Os estudantes também realizam um projeto prático, em grupos de 3 (três) estudantes. A avaliação do projeto tem em consideração o desempenho individual de cada estudante no grupo.*

*Tipo de Avaliação: Avaliação distribuída com exame final*

*Condições de Frequência:*

- *Presença mínima obrigatória nas aulas práticas: 75%*
- *Nota mínima em cada componente individual de avaliação:  $\geq 8,0$  valores*

*Fórmula de avaliação:*

- *Exame final (EF) sobre toda a matéria, com consulta, com duração de 2 horas*
- *Componente distribuída (CD), composta por um projeto sobre grafos*

*A nota final (NF) é dada por:  $NF = 0,6 * EF + 0,4 * CD$ , sujeito a EF e CD ser  $\geq 8,0$  valores*

**4.4.7. Teaching methodologies (including students' assessment):**

*Theory classes are structured so as to give students a formal exposition of the subject, alongside discussions on examples and their solutions. Lab classes are intended to provide students with tools and hands-on practice of algorithm development and their implementation (in C++, Java or Python) to test the devised solutions.*

*Students also develop a practical project, in groups of 3 students. Project evaluation takes into account the individual performance of each student in the group.*

*Type of evaluation: Distributed evaluation with final exam*

*Frequency Conditions:*

- *Minimum attendance in practical classes: 75%*
- *Minimum mark in each individual evaluation component:  $\geq 8.0$  (out of 20)*

*Evaluation formula:*

- *Final exam (FE) covering all subject, consultation allowed, in 2-hour duration time*
- *Distributed component (DC): a project on graphs*

*The final mark (FM) is given by:  $FM = 0.6 * FE + 0.4 * DC$ , subject to FE and DC  $\geq 8.0$  (out of 20)*

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

*A metodologia de ensino desta unidade curricular é caracterizada pela adoção de componentes teóricas e práticas de forma integrada, tanto nas aulas como nos vários momentos de avaliação. As aulas teóricas são usadas para a exposição formal da matéria, acompanhada da apresentação de exemplos e sua discussão. As aulas práticas são usadas para a resolução de exercícios e desenvolvimento de pequenos programas para testar os algoritmos desenvolvidos, com acompanhamento tutorial.*

*A consolidação prática do conhecimento teórico adquirido, numa perspectiva de “aprender fazendo” permite então ao estudante adquirir competências em: i) caracterizar um dado problema; ii) formalizar o problema de maneira precisa; e, iii) identificar a técnica de concepção de algoritmos mais apropriada para a sua solução. Numa perspectiva mais integrada e holística de todo o conhecimento adquirido, a realização de um projeto completo, como principal suporte da avaliação prática permite ao estudante consolidar as competências anteriormente mencionadas, e ainda iv) avaliar, tanto analítica assim como empiricamente a qualidade da solução concebida, a nível de eficiência e de correção. A partir do exame final, o estudante deve demonstrar autonomamente os conhecimentos adquiridos, tanto a nível prático assim como a nível teórico. Desta forma, os métodos de ensino e de avaliação propostos suportam a efetiva consolidação das quatro competências enunciadas anteriormente.*

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

*The teaching methodology of this course is characterized by the adoption of theoretical and practical components in an integrated manner, both in classes and in the various assessment components. The lectures are used for the formal exposition of the subject, accompanied by the presentation of examples and their discussion. Practical classes are used for solving exercises and developing small programs to test the algorithms developed with tutorial follow-up.*

*The practical consolidation of the acquired theoretical knowledge, in a “learning-by-doing” perspective, allows the student to acquire skills in: i) characterizing a given problem; ii) formalizing the problem precisely; and iii) identifying the most appropriate algorithm design technique to solve the problem at hand. In a rather integrated and holistic perspective of all the acquired knowledge, the implementation of a complete project as the main support of the practical evaluation of this course allows the student to consolidate the aforementioned competences, and iv) to evaluate, both analytically and empirically, the quality of the conceived solution, in terms of both efficiency and correction. Upon the final exam, the student must demonstrate autonomously the knowledge acquired, at both practical and theoretical levels. Thus, the proposed teaching and assessment methods support the effective consolidation of the four competences listed above.*

**4.4.9. Bibliografia de consulta/existência obrigatória:**

*Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). Cambridge, MA: MIT Press.*

*Weiss, M. A. (2014). Data Structures and Algorithm Analysis in C++ (4th ed). London: Pearson Education Ltd.*

*Skiena, S. S. (2008). The Algorithm Design Manual (2nd ed.). London: Springer.*

*Sedgewick, R. (2001). Algorithms in C++ Part 5: Graph Algorithms, (3rd ed.). Boston, MA: Addison-Wesley Professional.*

*Chapra, S. C., & Canale, R. P. (2010). Numerical methods for engineers. Boston: McGraw-Hill Higher Education.*

#### Mapa IV - Engenharia de Software

##### 4.4.1.1. Designação da unidade curricular:

*Engenharia de Software*

##### 4.4.1.1. Title of curricular unit:

*Software Engineering*

##### 4.4.1.2. Sigla da área científica em que se insere:

*EINFC*

##### 4.4.1.3. Duração:

*Semestral*

##### 4.4.1.4. Horas de trabalho:

*162*

##### 4.4.1.5. Horas de contacto:

*52 (26T + 26TP)*

##### 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):

*Ademar Manuel Teixeira de Aguiar (T - 26h - 2 turmas; TP - 26h - 10 turmas)*

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

*João Carlos Pascoal Faria (T – 26h – 2 turmas)*

*Filipe Alexandre Pais de Figueiredo Correia (TP - 78h - 10 turmas)*

*Hugo José Sereno Lopes Ferreira (TP - 104h - 10 turmas)*

*Docente a contratar (TP - 52h - 10 turmas)*

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

*Familiarizar-se com os métodos de engenharia e gestão necessários ao desenvolvimento de sistemas de software complexos e/ou em larga escala, de forma economicamente eficaz e com elevada qualidade.*

*No final da unidade curricular, os estudantes deverão ser capazes de:*

- 1. descrever os princípios, conceitos e práticas da engenharia de software e do ciclo de vida do software;*
- 2. conhecer e saber aplicar as técnicas e ferramentas necessárias para executar e gerir as várias atividades do processo de desenvolvimento de software de qualidade;*
- 3. explicar os métodos e processos de construção de diferentes tipos de sistemas de software.*

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

*This course aims to acquaint students with the engineering and management methods necessary for the cost-effective development and maintenance of high-quality complex software systems.*

*At the end of the semester, students should:*

- 1. be capable of describing the principles, concepts and practices of software engineering and software life cycle;*
- 2. be acquainted with and be capable of applying the required tools and techniques to carry out and manage the various tasks in the development of high-quality software;*
- 3. be capable of explaining the development methods and processes of different types of software systems.*

##### 4.4.5. Conteúdos programáticos:

*1. INTRODUÇÃO: objetivos, âmbito e história da engenharia de software.*

*2. PROCESSOS DE SOFTWARE: conceitos; modelos; atividades; processos tradicionais versus ágeis; RUP, Scrum, XP,*

OSS.

3. **GESTÃO DE PROJETOS DE SOFTWARE:** *estimação; planeamento, monitorização e controlo de projeto; gestão clássica versus gestão ágil.*

4. **ENGENHARIA DE REQUISITOS:** *conceitos; documentos; processo; casos de uso em UML; prototipagem de interfaces; user stories; cenários.*

5. **DESENHO DE SOFTWARE:** *arquitetura; modelação em UML; reutilização.*

6. **CONSTRUÇÃO DE SOFTWARE:** *ambientes; versões; integração contínua; desenvolvimento ágil com XP e OSS.*

7. **VERIFICAÇÃO E VALIDAÇÃO DE SOFTWARE:** *conceitos; testes unitários, integração, sistema e aceitação; BDD; revisões de software.*

8. **EVOLUÇÃO DE SOFTWARE:** *processos; dinâmicas; manutenção; sistemas legados; engenharia reversa e reengenharia do software.*

9. **MELHORIA DE PROCESSOS:** *ciclo de vida; métricas; CMMI; retrospectivas.*

#### 4.4.5. Syllabus:

1. **INTRODUCTION:** *objectives, scope and history of software engineering.*

2. **SOFTWARE PROCESS:** *concepts; models; activities; traditional methods versus agile methods; RUP, Scrum, XP, OSS.*

3. **SOFTWARE PROJECT MANAGEMENT:** *estimation; planning, monitoring and control; classic versus agile project management.*

4. **REQUIREMENTS ENGINEERING:** *concepts; documents; process; use cases and UML; user interface prototyping; user stories; scenarios.*

5. **SOFTWARE DESIGN:** *architecture; modeling with UML; reuse.*

6. **SOFTWARE CONSTRUCTION:** *environments; versioning; continuous integration; agile development with XP and OSS.*

7. **SOFTWARE VERIFICATION AND VALIDATION:** *concepts; unit, integration, system and acceptance testing; BDD; reviews.*

8. **SOFTWARE EVOLUTION:** *processes; dynamics; maintenance; legacy systems; reverse engineering and software re-engineering.*

9. **PROCESS IMPROVEMENT:** *lifecycle; measurement; CMMI; retrospectives.*

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

*Os conteúdos abordados na UC visam fornecer aos estudantes desde os conhecimentos fundamentais até às melhores práticas existentes, tradicionais e ágeis, na área de Engenharia de Software, bem como proporcionar a aplicação desses conhecimentos a um projeto concreto realizado em grupo ao longo de todo o semestre.*

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

*The contents of the course aim to provide the students with diverse knowledge, from the fundamentals to the best practices, traditional and agile, in the area of software engineering, as well as to enable students their application to a concrete project developed in group during all the semester.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*As aulas teóricas serão para exposição e discussão verbal, que incluem a descrição dos problemas, as metodologias de análise e as soluções e boas práticas preconizadas.*

*As aulas teórico-práticas serão para desenvolvimento de um projeto, em grupo, envolvendo um relatório de desenvolvimento e o software desenvolvido.*

*Tipo de Avaliação: Avaliação distribuída com exame final.*

*Condições de Frequência: Nota mínima de 40% em cada uma das componentes de avaliação. Não exceder o número limite de faltas e participar ativamente na elaboração dos trabalhos sujeitos a avaliação.*

*Os estudantes dispensados das regras de assiduidade devem, com periodicidade a combinar com os docentes, apresentar a evolução dos seus trabalhos, assim como participar na apresentação simultaneamente com os estudantes ordinários.*

*Fórmula de avaliação: Classificação Final = 65% Projeto + 35% Exame*

*Notas finais superiores ou iguais a 18 valores podem requerer uma prova oral.*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Theoretical classes will be based on the oral presentation of the themes and description of problems, as well methodology analysis and solutions/good practices.*

*Theoretical-practical classes will be used to develop a project, in groups, encompassing a development report and the software developed.*

*Type of evaluation: Distributed evaluation with final exam.*

*Terms of frequency: Minimum of 40% in each evaluation component. Not exceed the absence limits and actively participate in the practical works that will be subject to evaluation.*

*Students dispensed from attendance rules must, in time periods defined with teachers, present the evolution of their works, and participate in the final presentations sessions as the regular students.*

*Formula of evaluation: final Grade = 65% Project + 35% Exam*

*Final grades higher than or equal to 18 pts may require an oral examination.*

**4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular:**  
*As aulas são coerentes com os objetivos da unidade curricular. Em particular, a transmissão de conhecimentos nas aulas, o desenvolvimento de um projeto de software concreto e real onde os estudantes são orientados para aplicação das boas práticas ensinadas e o estudo para exame, vão permitir aos estudantes ganharem um conhecimento teórico e prático mais aprofundado sobre os tópicos da UC, bem como adquirirem capacidade crítica.*

**4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:**  
*Classes are consistent with the objectives of the course. In particular, the transmission of knowledge in the classroom, the development of a software project concrete and real where the students are supervised to apply the best practices lectured and the study for the exam, will help the students to acquire a solid theoretical and practical knowledge about the respective topics, as well to gain critical analysis skills on that topics.*

**4.4.9. Bibliografia de consulta/existência obrigatória:**  
*Sommerville, I. (2011). Software engineering 9th Edition. ISBN-10, 137035152.*  
*Kruchten. P. (2000). The Rational Unified Process: An Introduction, Second Edition (2nd. ed.). Addison-Wesley Longman Publishing Co., Inc., USA.*  
*Beck, K. & Andres, C. (2004). Extreme Programming Explained: Embrace Change (2nd Edition). Addison-Wesley Professional.*  
*Miles, R. & Hamilton, K. (2006). Learning UML 2.0. O'Reilly. ISBN: 0-596-00982-8.*  
*Silva, A. (2013). UML, metodologias e ferramentas CASE. ISBN: 989-615-009-5.*

#### Mapa IV - Física I

**4.4.1.1. Designação da unidade curricular:**  
*Física I*

**4.4.1.1. Title of curricular unit:**  
*Physics I*

**4.4.1.2. Sigla da área científica em que se insere:**  
*FIS*

**4.4.1.3. Duração:**  
*Semestral*

**4.4.1.4. Horas de trabalho:**  
*162*

**4.4.1.5. Horas de contacto:**  
*52 (26T + 26TP)*

**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):**  
*Jaime Enrique Villate Matiz (T - 52h - 2 turmas; TP - 78h - 12 turmas)*

**4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:**  
*Augusto da Silveira Rodrigues (TP - 52h - 12 turmas)*  
*Joana Cacilda Rodrigues Espain Oliveira (TP - 26h - 12 turmas)*  
*Luís Miguel Fortuna Rodrigues Martelo (TP - 78h - 12 turmas)*  
*Manuel António Salgueiro da Silva (TP - 52h - 12 turmas)*  
*Maria José Fernandes Vaz Lourenço Marques (TP - 26h - 12 turmas)*

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

*O objetivo desta unidade curricular é dar ao estudante conhecimentos básicos de mecânica, eletricidade e circuitos elétricos.*

*Após frequentar esta unidade curricular, os estudantes deverão adquirir as seguintes aptidões e competências:*

- 1. Resolução de equações de movimento simples, usando o método de separação de variáveis.*
- 2. Identificação das forças e torques que atuam num sistema mecânico e obtenção das equações de movimento.*
- 3. Análise de circuitos elétricos simples e explicação do seu funcionamento.*
- 4. Reconhecimento de fenómenos eletromagnéticos na experiência quotidiana.*
- 5. Uso de princípios físicos para explicar o funcionamento de aparelhos elétricos.*
- 6. Uso de sistemas de computação algébrica (CAS) na resolução de problemas de física.*

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

*This course aims to give the student basic background knowledge on mechanics, electricity and electrical circuits.*

*After completing this course, students should acquire the following skills and competences:*

- 1. Solution of simple equations of motion analytically, using the separation of variables method.*
- 2. Identification of the forces and torques acting on a mechanical system and derivation of the equations of motion.*
- 3. Analysis of simple electrical circuits and explanation of their working principles.*
- 4. Identification electromagnetic phenomena in the every day activities.*
- 5. Use of physical principles to explain how electric appliances work.*
- 6. Solution of physical problems using Computer Algebra Systems (CAS).*

#### **4.4.5. Conteúdos programáticos:**

- 1. Sistemas de Álgebra Computacional (CAS).*
- 2. Cinemática. Resolução das equações de movimento. Movimento em 3 dimensões.*
- 3. Movimentos dependentes. Movimento curvilíneo. Aceleração centrípeta.*
- 4. Dinâmica. Leis de Newton. Forças gravítica, elétrica e magnética.*
- 5. Forças de contacto. Atrito e resistência nos fluidos.*
- 6. Trabalho e energia. Conservação da energia.*
- 7. Movimento dos corpos rígidos. Momento de uma força e momento de inércia.*
- 8. Potencial e corrente elétrica. Fontes de força eletromotriz. Potência elétrica.*
- 9. Resistência elétrica. Capacidade elétrica e condensadores.*
- 10. Circuitos de corrente contínua, com fontes, resistências e condensadores.*

#### **4.4.5. Syllabus:**

- 1. Computer Algebra Systems (CAS).*
- 2. Kinematics. Solution of the motion equations. Motion in three dimensions.*
- 3. Dependent motions. Curvilinear motion. Centripetal acceleration.*
- 4. Dynamics. Newton laws. Gravitational, electrical and magnetic forces.*
- 5. Contact forces. Friction and fluid resistance.*
- 6. Work and energy. Energy conservation.*
- 7. Rigid bodies motion. Torque and moment of inertia.*
- 8. Potential and electric current. Electromotive-force sources. Electric power.*
- 9. Electric resistance. Electric capacity and capacitors.*
- 10. Direct-current circuits including sources, resistors and capacitors.*

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

*A mecânica é a base para a modelação de sistemas físicos abordada na unidade curricular Física 2. O processamento, armazenamento e transmissão de informação são feitos usando fenómenos eletromagnéticos. Consequentemente, a formação de base de um engenheiro informático deve incluir o estudo da eletricidade e o magnetismo. O programa proposto constitui a base para a compreensão dos fenómenos mecânicos e eletromagnéticos.*

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

*Mechanics is the basis for the modelling physical systems that will be the subject of the Physics 2 course. Information processing, storage and transmission are done using electromagnetic phenomena. Therefore, the background knowledge for a computer engineer must include the study of electricity and magnetism. The proposed syllabus constitutes the basis for an understanding of mechanical and electromagnetic phenomena.*

#### **4.4.7. Metodologias de ensino (avaliação incluída):**

*Nas aulas teórico-práticas os estudantes trabalham com computadores com acesso à Web e ao material disponibilizado que inclui apontamentos, perguntas de escolha múltipla e problemas. Nas aulas teóricas são realizadas demonstrações experimentais e são dados esclarecimentos sobre o material do livro de texto. O apoio à disciplina é feito através dum servidor de e-learning.*

*São realizados 3 minitests de uma hora durante o semestre, em datas marcadas no início do semestre. Cada miniteste abrange um terço dos capítulos do programa. Há um exame final, de recurso e para melhoria da classificação, sobre toda a matéria lecionada e com duração de duas horas.*

*Os testes e o exame são com consulta de um formulário e uso de um programa de computação algébrica (CAS). A nota final é a média das notas dos 3 testes, ou a nota do exame, no caso de melhoria.*

#### **4.4.7. Teaching methodologies (including students' assessment):**

*During the practical sessions students work in computers which access to the Web and to the support material including lecture notes, multiple-choice questions and proposed problems. The lectures are used to conduct experimental demonstrations and to explain the material on the textbook. The support for this course is done using an e-learning server.*

*Three one-hour quizzes are given during the semester, on dates which are fixed at the beginning of the course. The subject of each quiz is one third of the topics covered during the semester. A final exam is also given, to provide another chance for those who have not passed the course or those who would like to improved their grade; it covers all the topics of the course and it has a time-limit of two hours.*

*In the quizzes and the exams students can use an equation sheet and a Computer Algebra System (CAS). The final grade is the average of the grades of the three quizzes, or the exam grade in case of improvement.*

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

*Para que um estudante consiga analisar circuitos elétricos, reconhecer fenômenos eletromagnéticos na sua experiência quotidiana, usar princípios físicos para explicar o funcionamento dos aparelhos elétricos e avaliar diferentes dispositivos elétricos, é necessário que esteja familiarizado com os dispositivos elétricos e os aparelhos de medição elétrica.*

*Consequentemente nesta disciplina usa-se uma abordagem prática e um método de ensino ativo, com recurso a experiências simples e métodos computacionais. É usado um Sistema de Álgebra Computacional (CAS), para diminuir a barreira da falta do domínio da álgebra e a análise matemática constituem num curso de introdução à física.*

*Nas aulas teóricas são introduzidas as leis físicas que permitem explicar os fenômenos físicos estudados. Nas aulas teórico-práticas os estudantes podem explorar livremente os conteúdos disponibilizados e pesquisar na Web, discutindo com os seus colegas e com o docente. Os conteúdos disponibilizados incluem muitas perguntas conceituais e problemas. Esta metodologia enfatiza o compreensão dos fenômenos físicos que é o objetivo principal da unidade curricular.*

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

*To be able to analyze electrical circuits, identify electromagnetic phenomena in their daily experience, use physical principles to explain how electric appliances work and evaluate different electrical devices, students must be familiar with electrical devices and measuring instruments.*

*Hence, in this course we use a hands-on approach and an active teaching methodology, using simple experiments and computational methods. A Computer Algebra System (CAS) is used, in order to lower the barrier that a lack of expertise in algebra and mathematical analysis poses in an introductory physics course.*

*The master lectures are used to introduce the physical laws used to explain the physical phenomena being studied. In the practical sessions the students can freely explore the materials provided, and to search the Web, discussing with their colleagues and the teacher. The material provided for the course includes many conceptual questions and problems. This methodology emphasizes the understanding of physical phenomena, which is the main objective of the course.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Villate, J. E. (2016). Dinâmica e sistemas dinâmicos. Author's edition, Porto, Portugal. Distributed by Amazon.*

*Villate, J. E. (2015). Electricidade, magnetismo e circuitos. Author's edition, Porto, Portugal. Distributed by Amazon.*

*French, A. P. (1971). Newtonian Mechanics. New York, NY, USA: W. W. Norton & Company.*

*Tipler, P. A., & Mosca, G. (2004, 5ª edição). Physics. New York, NY, USA: W. H. Freeman and Co.*

*Alonso, M. & Finn, E. J. (1999). Física. Reading, MA, USA: Addison-Wesley.*

### **Mapa IV - Física II**

#### **4.4.1.1. Designação da unidade curricular:**

*Física II*

#### **4.4.1.1.1. Title of curricular unit:**

*Physics II*

#### **4.4.1.1.2. Sigla da área científica em que se insere:**

*FIS*

#### **4.4.1.1.3. Duração:**

*Semestral*

#### **4.4.1.1.4. Horas de trabalho:**

*121.5*

#### **4.4.1.1.5. Horas de contacto:**

*39 (19.5T + 19.5TP)*

#### **4.4.1.1.6. ECTS:**

## 4.5

**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):***Jaime Enrique Villate Matiz (T - 39h - 2 turmas; TP - 19.5h – 11 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***António José dos Santos Silva (TP - 39h - 11 turmas)**Ariel Ricardo Negrão da Silva Guerreiro (TP - 39h - 11 turmas)**Carla Susana Santana Carmelo Rosa (TP - 39h - 11 turmas)**Joana Cacilda Rodrigues Espain Oliveira (TP - 39h - 11 turmas)**Luís Miguel Fortuna Rodrigues Martelo (TP - 39h - 11 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***O objetivo desta unidade curricular é dar ao estudante conhecimentos básicos de modelação de sistemas físicos e métodos computacionais usados para resolver sistemas dinâmicos.**Após frequentar esta unidade curricular, os estudantes deverão adquirir as seguintes aptidões e competências:*

- 1. Descrição das forças elementares em termos de campos vetoriais e campos escalares.*
- 2. Caracterização dos diferentes tipos de ondas eletromagnéticas.*
- 3. Uso de campos de direções na resolução de sistemas de equações diferenciais.*
- 4. Análise dum sistema dinâmico identificando o tipo de sistema, variáveis de estado e equações de evolução.*
- 5. Identificação e caracterização dos pontos de equilíbrio de um sistema dinâmico.*
- 6. Resolução numérica das equações de evolução de um sistema dinâmico e interpretação das soluções obtidas.*

**4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***This course aims to give the student basic knowledge on modeling of physical systems and computational techniques used to solve dynamical systems.**After completing this course, students should acquire the following skills and competences:*

- 1. Description of the elementary forces in terms of vector and scalar fields.*
- 2. Characterization of the different types of electromagnetic waves.*
- 3. Use of direction fields to solve systems of differential equations.*
- 4. Analysis of a dynamical system identifying its state variables, evolution equations and type of system.*
- 5. Finding and characterizing the equilibrium points of a dynamical system.*
- 6. Numerical solution of the evolution equations of a dynamical system and interpretation of the obtained solutions.*

**4.4.5. Conteúdos programáticos:**

- 1. Campos elétrico e magnético. Fluxo elétrico e magnético.*
- 2. Indução eletromagnética. Indutância e autoindução.*
- 3. Ondas eletromagnéticas e luz. Espetro eletromagnético. Teorias ondulatória e corpuscular da luz.*
- 4. Sistemas dinâmicos. Espaço de fase. Equilíbrio estável e instável.*
- 5. Sistemas lineares. Osciladores harmónicos. Análise de estabilidade.*
- 6. Sistemas não lineares. Aproximação linear. Resolução numérica de equações diferenciais.*
- 7. Ciclos limite e sistemas de duas espécies. Sistemas predador presa.*
- 8. Sistemas caóticos. Comportamento assintótico, atratores estranhos e sistemas caóticos.*

**4.4.5. Syllabus:**

- 1. Electric and magnetic fields. Electric and magnetic flux.*
- 2. Electromagnetic induction. Inductance and self-induction.*
- 3. Electromagnetic waves and light. Electromagnetic spectrum. Ondulatory and corpuscular theories of light.*
- 4. Dynamical systems. State space. Stable and unstable equilibrium.*
- 5. Linear systems. Harmonic oscillators. Stability analysis.*
- 6. Non linear systems. Linear approximations. Numerical solution of differential equations.*
- 7. Limit cycles and two-species systems. Predator-prey systems.*
- 8. Chaotic systems. Asymptotic behavior, strange attractors and chaotic systems.*

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

*A Física é uma das ciências de base de qualquer Engenharia. Com o desenvolvimento dos computadores pessoais, o tipo de problemas físicos que podem ser resolvidos numa disciplina introdutória aumentou significativamente. Consequentemente, a formação de base de um engenheiro informático deve incluir modelação de sistemas físicos e métodos computacionais para resolução de problemas nessa área. O programa proposto constitui a base para a compreensão desses métodos.*

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

*Physics is the foundation of any Engineering field. With the advent of computers, the kind of physical problems that can be solved in an introductory course has expanded significantly. Therefore, the background knowledge for a computer engineer must include modelling of physical systems and computational methods to solve problems in that area. The proposed syllabus constitutes the basis for an understanding of those methods.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*Nas aulas teórico-práticas os estudantes trabalham com computadores com acesso à Web e ao material disponibilizado que inclui apontamentos, perguntas de escolha múltipla e problemas. Nas aulas teóricas são realizadas demonstrações experimentais e são dados esclarecimentos sobre o material do livro de texto. O apoio à disciplina é feito através dum servidor de e-learning.*

*São realizados 2 minitests de uma hora durante o semestre, em datas marcadas no início do semestre e um exame final de duas horas. Cada miniteste abrange metade dos capítulos do programa e o exame é sobre toda a matéria lecionada. Há um segundo exame, de recurso e para melhoria da classificação.*

*Os testes e os exames são com consulta de um formulário e uso de um programa de computação algébrica (CAS). Sendo D a nota dos testes e E a nota do exame, a nota final calcula-se com a fórmula seguinte:*

*Máximo ( E;  $0.4 \cdot D + 0.6 \cdot E$  )*

**4.4.7. Teaching methodologies (including students' assessment):**

*During the practical sessions students work in computers which access to the Web and to the support material including lecture notes, multiple-choice questions and proposed problems. The lectures are used to conduct experimental demonstrations and to explain the material on the textbook. The support for this course is done using an e-learning server.*

*Two one-hour quizzes are given during the semester, on dates which are fixed at the beginning of the course, and there is a final exam at the end. The subject of each quiz is one half of the topics covered during the semester, while the exam covers all the topics of the course. A second exam is also given, to provide another chance for those who have not passed the course or those who would like to improved their grade.*

*The final exam which has a time limit of 2 hours covers all the topics covered.*

*Being D the grade of the quizzes and E the exam grade, the final grade is calculated with the following equation:*

*Maximum ( E;  $0.4 \cdot D + 0.6 \cdot E$  )*

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

*A Física Computacional e as técnicas de simulação permitem que o aluno possa ter uma visão geral de um problema de física, sem ter que usar técnicas analíticas complicadas. As técnicas computacionais desenvolvidas para resolver problemas de mecânica têm sido aplicadas com sucesso em outros campos fora da física, dando origem à teoria geral dos sistemas dinâmicos.*

*Consequentemente nesta disciplina usa-se uma abordagem prática e um método de ensino ativo, com recurso a experiências simples e métodos computacionais. É usado um Sistema de Álgebra Computacional (CAS), para permitir que o estudante possa modelar sistemas físicos sem perder muito tempo em aprender métodos abstratos.*

*Nas aulas teóricas são introduzidas as leis físicas que permitem explicar os fenómenos físicos estudados. Nas aulas teórico-práticas os estudantes podem explorar livremente os conteúdos disponibilizados, e pesquisar na Web, discutindo com os seus colegas e com o docente. Os conteúdos disponibilizados incluem muitas perguntas conceituais e problemas. Esta metodologia enfatiza o compreensão dos fenómenos físicos que é o objetivo principal da unidade curricular.*

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

*Computational physics and simulation techniques allow students to get a wider view of physical phenomena without requiring complicated analytical methods. The computational techniques developed to solve mechanics problems have been applied to other areas outside physics, giving rise to the general theory of dynamical systems.*

*Hence, in this course we use a hands-on approach and an active teaching methodology, using simple experiments and computational methods. A Computer Algebra System (CAS) is used, in order to allow the student to model physical systems without spending much time learning abstract analytical techniques.*

*The master lectures are used to introduce the physical laws used to explain the physical phenomena being studied. In the practical sessions the students can freely explore the materials provided, and to search the Web, discussing with their colleagues and the teacher. The material provided for the course includes many conceptual questions and problems. This methodology emphasizes the understanding of physical phenomena, which is the main objective of the course.*

**4.4.9. Bibliografia de consulta/existência obrigatória:**

*Villate, J. E. (2016). Dinâmica e sistemas dinâmicos. Author's edition, Porto, Portugal. Distributed by Amazon.*

*Villate, J. E. (2015). Electricidade, magnetismo e circuitos. Author's edition, Porto, Portugal. Distributed by Amazon.*

*Strogatz, S. H. (2000). Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering. Cambridge, MA, USA: Perseus Books.*

*Tipler, P. A., & Mosca, G. (2004, 5ª edição). Physics. New York, NY, USA: W. H. Freeman and Co.*

*Alonso, M. & Finn, E. J. (1999). Física. Reading, MA, USA: Addison-Wesley.*

**4.4.1.1. Designação da unidade curricular:***Fundamentos da Programação***4.4.1.1. Title of curricular unit:***Programming Fundamentals***4.4.1.2. Sigla da área científica em que se insere:***EINFC***4.4.1.3. Duração:***Semestral***4.4.1.4. Horas de trabalho:***162***4.4.1.5. Horas de contacto:***52 (26T + 26TP)***4.4.1.6. ECTS:***6***4.4.1.7. Observações:***<sem resposta>***4.4.1.7. Observations:***<no answer>***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***João António Correia Lopes (T - 26h - 2 turmas; TP - 52h - 12 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***João Pedro Pedroso Ramos Santos (T - 26h - 2 turmas; TP - 52h - 12 turmas)**Fernando José Cassola Marques (TP - 52h - 12 turmas)**Luis Filipe Pinto de Almeida Teixeira (TP - 52h - 12 turmas)**Docente a contratar (TP - 104h - 12 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***O objetivo desta Unidade é dar ao estudante a capacidade de criar algoritmos e de usar uma linguagem de programação para implementar, testar e depurar algoritmos para resolver problemas simples.**O estudante será capaz de entender e usar os conceitos fundamentais de programação e a abordagem funcional da programação, especificamente a programação livre de efeitos (effect-free), onde as chamadas de função não têm efeitos colaterais e as variáveis são imutáveis, e de contrastar esta abordagem com a abordagem imperativa.**No final da unidade, espera-se que o estudante consiga resolver problemas através do desenvolvimento de programas de complexidade média, usando as abordagens ou paradigmas de programação imperativa e funcional.***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***The aim of this Unit is to give the student the ability to create algorithms, and to use a programming language to implement, test, and debug algorithms for solving simple problems.**The student will be able to understand and use the fundamental programming constructs, and the functional approach to programming, specifically effect-free programming where function calls have no side-effects and variables are immutable, and contrast it with the Imperative approach.**At the end of the course, the student is expected to handle programming problems of medium complexity, using the imperative or functional programming approaches or paradigms.***4.4.5. Conteúdos programáticos:***1. Introdução ao Pensamento Computacional, algoritmos e programação com Python.**2. Conceitos fundamentais de programação: tipos de dados simples; variáveis, expressões e declarações; fluxo de programa, condicionais, iteração; funções, passagem de parâmetros, recursão; persistência.**3. Tipos de dados compostos: strings, tuplos, conjuntos, listas, dicionários.**4. Programação livre de efeitos usando chamadas a função sem efeitos colaterais e variáveis imutáveis.**5. Funções como cidadãos de primeira classe e closures.**6. Funções de ordem superior: map, reduce/fold e filter.**7. Geradores e compreensões em listas.*

8. *Estratégias de resolução de problemas.*
9. *Ferramentas de programação, teste e depuração.*

#### 4.4.5. Syllabus:

1. *Introduction to Computational Thinking, algorithms and programming with Python.*
2. *Fundamental programming concepts: simple data types; variables, expressions and statements; program flow, conditionals, iteration; functions, parameter passing, recursion; persistence; functions, parameter passing and recursion*
3. *Working with aggregate data types: strings, tuples, sets, lists, dictionaries.*
4. *Effect-free programming by using function calls without side effects and immutable variables.*
5. *First-class functions and closures.*
6. *Utility higher-order functions: map, reduce/fold and filter.*
7. *Generators and list comprehensions.*
8. *Problem-solving strategies.*
9. *Programming tools, debugging, exceptions.*

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

*O item 1 dos Conteúdos Programáticos cobre o item 1 dos Objetivos (1 -> 1); 2 -> 2, 3, 4; 3 -> 2; 4 -> 5, 6; 5 -> 5, 6; 6 -> 7; 7 -> 7; 8 -> 1; 9 -> 8.*

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

*Item 1 of the Syllabus covers item 1 of the Learning Outcomes (1 -> 1); 2 -> 2, 3, 4; 3 -> 2; 4-> 5,6; 5 -> 5, 6; 6 -> 7; 7 -> 7; 8 -> 1; 9 -> 8.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*Tipo de Avaliação: Avaliação distribuída sem exame final*

*Fórmula de avaliação: A avaliação será baseada nas seguintes componentes:*

*- LE = Perguntas de resposta múltipla, sobre conceitos de programação, realizadas individualmente no Moodle em sala de aula teórica*

*- RE = Exercícios de programação semanais para casa e nas aulas teórico-práticas*

*- PE = Avaliação prática em computador a realizar, individualmente, no Moodle*

*- TE = Avaliação teórica através de questionário de respostas múltiplas, a realizar individualmente no Moodle com consulta de um livro*

*Classificação final = 10% LE + 10% RE + 50% PE + 30% TE*

*Observações:*

*1. É exigida classificação mínima de 40% na componente TE*

*2. É exigida a classificação mínima de 40% em pelo menos uma das 2 últimas provas (PE04 ou PE05)*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Type of evaluation: Distributed evaluation without final exam*

*Formula of evaluation: The evaluation will be based on the following components:*

*- LE = Lecture in-class evaluation: Multiple-choice questions about programming concepts, to be answered on an individual basis in the theoretical classroom*

*- RE = Recitation and away weekly evaluation: Weekly recitation classes and away programming assignments*

*- PE = Practical on computer evaluation: Individual programming assignments in Moodle*

*- TE = Theory evaluation: Multiple-choice questions about programming concepts, to be answered on an individual basis in Moodle with the consultation of one book*

*Final classification = 10% LE + 10% RE + 50% PE + 30% TE*

*Observations:*

*1. A minimum classification of 40% in the component TE is required.*

*2. A minimum mark of 40% is required for at least one of the last two practical on computer evaluation assessments (PE04 or PE05).*

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

*O envolvimento contínuo do estudante com a unidade é promovido, através do estudo e discussão dos tópicos, distribuídos previamente em Notebooks Jupyter, tanto em aulas teóricas e teórico-práticas, como através de trabalhos de programação dentro e fora das aulas.*

*O estudante é encorajado a encontrar as melhores ideias para resolver problemas específicos, executá-las e implementar as soluções de programação, de forma elegante, legível e eficiente (em tempo e em espaço) usando a linguagem de programação Python.*

*São usadas ferramentas de correção automática de código para aumentar a rapidez do feedback dado aos estudantes. As aulas teóricas (T) são usadas para apresentar e discutir os tópicos do programa, usando um computador ligado a um projetor multimédia.*

*As aulas teórico-práticas (TP) são usadas para ajudar os estudantes a entender os tópicos do programa e a resolver as tarefas de programação semanais.*

*As tarefas de programação em aula teórica e fora das aulas são dadas, semanalmente, para melhorar o desenvolvimento regular e eficaz dos processos de aprendizagem individual autónomo e são avaliadas em tarefas do Moodle.*

*Tarefas em sala de aula são usadas para avaliação somativa no final de cada aula teórica através de questionários no Moodle.*

*Os estudantes são incentivados a usar uma App Web (Play) contendo exercícios de programação selecionados por tema: exercícios (fáceis) para realizar antes da aula teórica, exercícios (dificuldade média) para realizar antes das tarefas semanais (RE) e exercícios (difíceis) para realizar antes das provas práticas em computador (PE).*

*No trabalho em sala de aula, os estudantes usam o mesmo ambiente de trabalho (IDE Spyder, Pylint, App Web (Test) e de submissão de exercícios de programação) que é usado posteriormente nas avaliações individuais (PE).*

*Sempre que for considerado necessário na sala de aula, especialmente durante o primeiro mês de aulas da unidade, os estudantes são incentivados a passar pela "Clínica", assegurada por monitores, para obter ajuda.*

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

*The continuous enrollment of the student in the course is promoted, through the study and discussion of the course topics, previously distributed in Jupyter Notebooks, both in lectures and recitation classes and with in-class and away programming assignments.*

*The student is motivated to find the best ideas to solve specific problems, execute them and implement the programming solutions, in an elegant, legible and efficient (time and space) mode using the Python programming language.*

*Automatic correction tools are used to increase the efficiency of the feedback given to the students.*

*Lecture classes (T) are used to present and discuss the topics of the program, using a computer connected to a multimedia projector.*

*Recitation classes (TP) are used to help students understand the topics of the program and solve the weekly programming assignments.*

*In-class and away programming assignments are given on a weekly basis, to improve the regular and effective development of autonomous learning processes and are tested and graded using Moodle assignments.*

*In-class assignments are used for summative evaluation at the end of each lecture class using Moodle quizzes.*

*Students are encouraged to use a Web App (Play) containing programming exercises selected by theme: exercises (easy) to perform before the theoretical class, exercises (of average difficulty) to perform before the weekly assignments (RE) and exercises (difficult) to perform before the practical tests in computer (PE).*

*In the classroom, the students use the same working environment (IDE Spyder, Pylint, Web App (Test) and submissions of weekly assignments) that is used later in the individual assessments (PE).*

*Students are encouraged to go to the "Clinic", which is supervised by monitors, in their first weeks of classes and whenever necessary in the classroom, especially during the first month.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Downey, A.B., Elkner, J., Wentworth, P., & Meyers, C. (2019). How to Think Like a Computer Scientist: Learning with Python 3 (GNU Free Documentation License). ReadTheDocs.org (eBook).*

*Lott, S. F. (2010). Building Skills in Python: A Programmer's Introduction to Python. itmaybeahack.com (ebook).*

*Mertz, D. (2015). Functional Programming in Python. O'Reilly Media.*

*Costa, E. (2015). Programação em Python - Fundamentos e Resolução de Problemas (FCA Editora, ed.).*

### **Mapa IV - Fundamentos de Segurança Informática**

#### **4.4.1.1. Designação da unidade curricular:**

*Fundamentos de Segurança Informática*

#### **4.4.1.1. Title of curricular unit:**

*Computer Security Foundations*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

*52 (26T + 26TP)*

#### **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):***Manuel Bernardo Martins Barbosa (T - 26h - 2 turmas; TP - 52h - 12 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***António Miguel Pontes Pimenta Monteiro (T - 26h - 2 turmas; TP - 52h - 12 turmas)**João Paulo da Silva Machado Garcia Vilela (TP - 52h - 12 turmas)**José Manuel de Magalhães Cruz (TP - 52h - 12 turmas)**Luís Filipe Coelho Antunes (TP - 52h - 12 turmas)**Manuel Eduardo Carvalho Duarte Correia (TP - 52h - 12 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Conhecer os princípios da construção de programas e sistemas informáticos seguros.**Aprender a pensar sobre sistemas informáticos como um atacante.**Compreender como se identificam ameaças a um sistema informático e como se avalia a sua relevância.**Reconhecer limitações e justificar medidas de proteção de um sistema informático.**Explicar como diversos tipos de ataques funcionam na prática.***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***To know the principles of building secure programs and computer systems.**To learn how to think adversarially about computer systems.**To understand how to assess threats for their significance.**To recognize limitations and justify protections of a given computer system.**To explain how attacks work in practice.***4.4.5. Conteúdos programáticos:***1) Princípios da segurança informática: confidencialidade, integridade, disponibilidade; risco, ameaças, vulnerabilidades, vetores de ataque e mecanismos de segurança .**2) Princípios da construção de sistemas seguros: princípios do privilégio mínimo e isolamento; defesa em profundidade; segurança por construção.**3) Conceitos básicos de criptografia: criptografia simétrica e de chave pública; cifras e autenticação; assinaturas digitais; gestão de chaves; PKI.**4) Controlo de acessos: conceitos fundamentais; modelos para controlo de fluxos de dados; mecanismos de segurança ao nível do sistema operativo.**5) Introdução à programação defensiva: validação de inputs; ataques comuns; buffer overflows; race conditions; atualizações de segurança.**6) Tópicos de segurança de redes: ataques e proteção ao nível da rede; ataques por Denial of Service (DoS) e Distributed Denial of Service (DDoS).**7) Segurança Web: modelo de segurança; gestão de sessões; autenticação; vulnerabilidades comuns.***4.4.5. Syllabus:***1) Principles of computer security: confidentiality, integrity, availability; risk, threats, vulnerabilities, attack vectors, security mechanisms.**2) Principles of secure design: least privilege and isolation; defense in depth; security by design.**3) Basic cryptography concepts: symmetric and public-key cryptography; hash functions; encryption and authentication; digital signatures; key management; PKI.**4) Access control: basic concepts; information-flow control and models for confidentiality and integrity; security mechanisms at the OS level.**5) Introduction to defensive programming: input validation; common vulnerabilities and attacks; buffer overflows; race conditions; security updates.**6) Topics in network security: attacks and protection at the network level; Denial of Service (DoS) and Distributed Denial of Service (DDoS).**7) Web security: security model; session management; authentication; common vulnerabilities.***4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:***Esta Unidade Curricular tem como objetivo fornecer aos estudantes uma perspectiva integrada dos fundamentos da segurança informática; visa dotar os estudantes de uma visão abrangente dos aspetos de segurança inerentes ao desenvolvimento e operação de sistemas informáticos, contextualizando problemas e soluções tecnológicas específicas para tecnologias abordadas em outras Unidades Curriculares do Ciclo de Estudos. Os conteúdos programáticos estão alinhados com as recomendações dos ACM Computer Science Curricula para o 1o Ciclo na área da Engenharia Informática e os objetivos de aprendizagem focam as competências fundamentais necessárias à integração de profissionais nesta área em equipas de desenvolvimento cujas atividades incluam a análise, conceção, implementação e validação de propriedades de segurança em sistemas informáticos.***4.4.6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes:**

*The goal of this Curricular Unit is to provide students with an integrated perspective of the foundations of computer security; it aims to give students a broad view of the security aspects inherent to the development and operation of computer systems, setting a context for the technology-specific problems and solutions students encounter in other Curricular Units. The syllabus is aligned with the ACM Computer Science Curricula for undergraduate degrees in Computer Science and the learning outcomes focus on the fundamental skills that professionals in this area require to productively participate in development teams that handle the analysis, design, implementation and validation of security properties in computer systems.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*As aulas teóricas são de exposição, apresentadas normalmente com auxílio de projector. Ao longo do semestre serão propostos problemas de programação e outros para consolidação dos conceitos apresentados nas aulas teóricas. Estes problemas serão discutidos nas aulas teórico-práticas, mas espera-se que os estudantes os resolvam fora das aulas. Os estudantes deverão ainda realizar um projeto em grupo, onde irão aplicar os conceitos apresentados nas aulas teóricas. Os aspetos práticos serão também aprofundados através da articulação com Unidades Curriculares que os estudantes frequentam em simultâneo, nomeadamente redes de computadores.*

*Avaliação distribuída com exame final.*

*A fórmula de cálculo da classificação final é:*

$$0,4 PP + 0,6ET$$

*onde:*

*PP - Nota do projecto*

*ET - Classificação do Exame Teórico*

#### 4.4.7. Teaching methodologies (including students' assessment):

*The lectures are based on oral presentation, complemented with detailed examples and the discussion of case studies. Consolidation exercises will be proposed during the semester; these will be discussed in lectures but it is expected that students complete them outside of class. Students will also develop a group project, where they will apply the concepts covered in class. Practical aspects will be further developed in articulation with Curricular Units that the students attend simultaneously, namely computer networks.*

*Distributed assessment with a final written exam.*

*The formula for computation of the final mark is:*

$$0,4 PP + 0,6ET$$

*where:*

*PP – Project mark*

*ET – Written exam mark*

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

*As aulas expositivas fornecem uma visão coerente e integrada dos tópicos selecionados, com uma ligação permanente a exemplos práticos que estabelecem uma ligação entre os conceitos apresentados e uma visão detalhada da sua concretização em sistemas informáticos reais.*

*A discussão dos casos de estudo e a resolução de exercícios permite a consolidação dos conhecimentos e aprofundar a análise e reflexão crítica, criando-se através do projeto também oportunidades para trabalho colaborativo onde se promove o desenvolvimento de competências de comunicação técnica no contexto do tratamento de aspetos relacionados com a segurança informática.*

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

*The exposition in class will provide a coherent an integrated view of the selected topics with a constant connection to practical examples that link the presented concepts to a detailed view of how they occur in real-world computer systems.*

*The discussion of case studies and problem solving allows students to consolidate learning and to carry out more in-depth analysis and critical thought; the group project also creates opportunities for collaborative work where technical communication abilities in the context of computer security can be exercised.*

#### 4.4.9. Bibliografia de consulta/existência obrigatória:

*Goodrich, M., & Tamassia, R. (2014). Introduction to Computer Security. Pearson. ISBN: 978-0321512949.*

*Gollmann, D. (2010). Computer Security. Wiley. ISBN: 978-0133575477.*

### Mapa IV - Fundamentos de Sistemas Computacionais

#### 4.4.1.1. Designação da unidade curricular:

*Fundamentos de Sistemas Computacionais*

#### 4.4.1.1. Title of curricular unit:

*Fundamentals of Computer Systems*

#### 4.4.1.2. Sigla da área científica em que se insere:

**EINFC****4.4.1.3. Duração:***Semestral***4.4.1.4. Horas de trabalho:***162***4.4.1.5. Horas de contacto:***52 (26T + 26TP)***4.4.1.6. ECTS:***6***4.4.1.7. Observações:***<sem resposta>***4.4.1.7. Observations:***<no answer>***4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***João Paulo de Castro Canas Ferreira (T - 26h - 2 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Luís Miguel Barros Lopes (T - 26h - 2 turmas; TP - 78h - 12 turmas)**António José Duarte Araújo (TP - 78h - 12 turmas)**Bruno Miguel Carvalhido Lima (TP - 52h - 12 turmas)**João Paulo Filipe de Sousa (TP - 52h - 12 turmas)**Docente a contratar (TP - 52h - 12 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):**

- 1. Descrever os principais subsistemas de um computador;*
- 2. Descrever e interpretar formatos numéricos básicos;*
- 3. Efetuar operações aritméticas em binário;*
- 4. Manipular expressões de álgebra de Boole;*
- 5. Definir e usar circuitos lógicos combinatórios e sequenciais simples;*
- 6. Explicar a funcionalidade de circuitos lógicos padrão;*
- 7. Explicar o modelo conceptual de funcionamento de memórias de estado sólido e analisar o processo de descodificação de endereços;*
- 8. Explicar os princípios básicos da codificação de instruções e modos de endereçamento (arquitetura do conjunto de instruções);*
- 9. Escrever programas simples em linguagem “assembly” com operações aritméticas e booleanas, testes, saltos e sub-rotinas; gestão de pilha.*
- 10. Descrever o funcionamento de uma unidade de processamento simples;*
- 11. Avaliar o desempenho de processadores em cenários simples.*

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

- 1. Identify and describe the major subsystems of a personal computer;*
- 2. Describe and interpret binary representation of numerical information;*
- 3. Perform basic arithmetic operations in binary;*
- 4. Handle symbolic switching algebra expressions;*
- 5. Identify and explain the operation of basic logic circuits (combinational and sequential);*
- 6. Explain the operation of standard logic modules;*
- 7. Explain the conceptual operation of solid-state memories and the process of address decoding;*
- 8. Explain the basic principles of instruction encoding and addressing modes (instruction set architecture);*
- 9. Write simple programs in assembly language involving Boolean and arithmetic operations, tests, jumps and subroutines;*
- 10. Describe the operation of a single-cycle processing unit.*
- 11. Evaluate the performance of processors in simple scenarios.*

**4.4.5. Conteúdos programáticos:***M1. Introdução: Áreas de aplicação de computadores e suas características.**M2. Representação de informação: Representação binária de números inteiros. Operações aritméticas elementares. Códigos. Vírgula flutuante.**M3. Circuitos combinatórios: Álgebra de Boole. Portas lógicas elementares. Diagramas lógicos. Simulador lógico. Circuitos com funções padrão.*

- M4. Circuitos sequenciais: Elementos de memória; Registos e contadores; Descodificação de endereços.  
 M5. Computadores: Linguagens de alto e baixo nível. Modelo concetual da execução de um programa. Subsistemas: CPU, memória, periféricos.  
 M6. Conjunto de instruções: Tipos de instruções, modos de endereçamento, codificação.  
 M7. Linguagem “assembly”: Conceitos básicos de programação.  
 M8. Implementação de controlo de fluxo, incluindo sub-rotinas; gestão da pilha.  
 M9. Unidade de processamento: Organização lógica de um CPU simples (de ciclo único).  
 M10. Desempenho: Equação básica. Benchmarks. Lei de Amdahl.

#### 4.4.5. Syllabus:

- M1. Introduction. Computers: application areas of and their characteristics.  
 M2. Representation of information: binary representation of integers. Elementary arithmetic operations. Codes. IEEE-754 floating-point format.  
 M3. Combinational logic circuits. Boolean expressions. Elementary logic gates. Logic diagrams. Logic simulator. Standard circuits.  
 M4. Synchronous circuits: Memory elements, registers and counters. Address decoding.  
 M5. Computers: high- and low-level languages. Conceptual model of program execution. Subsystems: CPU, memory, input/output peripherals.  
 M6. Instruction set: Instruction types, addressing modes, instruction encoding.  
 M7. Basic concepts of assembly programming.  
 M8. Implementation of control flow structures, including procedures; stack management.  
 M9. Organization of a simple central processing unit. Single-cycle CPU; limitations.  
 M10. CPU performance: Basic performance equation, benchmarks, Amdahl's Law.

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

A correspondência entre conteúdos programáticos e os objetivos de aprendizagem é quase direta, conforme descrito a seguir.

- M1: Objetivo 1.  
 M2: Objetivos 2 e 3.  
 M3: Objetivos 4, 5 e 6.  
 M4: Objetivos 5, 6, e 7.  
 M5: Objetivo 1, revisitado a um maior nível de profundidade (empregando conceitos de sistemas lógicos).  
 M6: Objetivo 8.  
 M7: Objetivo 9.  
 M8: Objetivo 9.  
 M9: Objetivos 10.  
 M10: Objetivo 11.

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

The syllabus maps onto the learning objectives in a rather straightforward way, as described next:

- M1: Objective 1.  
 M2: Objective 2 e 3.  
 M3: Objectives 4, 5 e 6.  
 M4: Objectives 5, 6, e 7.  
 M5: Objective 1, revisited at a deeper level (applying concepts of logic design).  
 M6: Objective 8.  
 M7: Objective 9.  
 M8: Objective 9.  
 M9: Objective 10.  
 M10: Objective 11.

#### 4.4.7. Metodologias de ensino (avaliação incluída):

A unidade curricular tem uma componente teórica baseada em aulas de exposição dos diversos temas acompanhados da apresentação de exemplos e respetiva discussão. As aulas teórico-práticas incluem a apresentação, análise e resolução de um conjunto de questões e de casos de estudo. Exercícios de programação serão testados num emulador de CPU.

Atividades de aprendizagem a realizar fora do período de aulas: Questionários de escolha múltipla (autoavaliação).

Avaliação: distribuída sem exame final.

A avaliação é composta por dois testes (2h cada), com a nota final dada pela respetiva média.

Prova de repescagem destinada exclusivamente a estudantes que obtiverem nota final inferior a 10 valores. Cada teste com nota inferior a 9,5 pode ser repescada de forma independente. Em cada teste de repescagem a nota máxima atribuída é de 9,5 valores (em 20).

#### 4.4.7. Teaching methodologies (including students' assessment):

The course includes lectures on the subject matter with the presentation of examples and their discussion. The theoretical-practical sessions include the presentation, analysis and resolution of a number of problems. Programming exercises will be tested on CPU emulator.

Learning activities outside the classroom: Multiple-choice questionnaires for self-assessment.

Type of evaluation: Distributed evaluation without final exam

The course grade is calculated as the average of two tests (2h each).

*There will be extra tests exclusively for students who obtain a final score lower than 10. Each test with score below 9.5 may be retaken once. The maximum score awarded for the extra test is 9.5 (out of 20) (for each test).*

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

*O primeiro teste avalia os conteúdos programáticos abordados nos módulos M1-M4 (essencialmente, sistemas digitais) e o segundo teste os conteúdos dos módulos M5-M10 (processador e memória). Trata-se de uma divisão natural do conteúdo, que fica totalmente coberto pelos dois momentos de avaliação.*

*Os testes incluem tanto questões de escolha múltipla como questões de resposta aberta que permitem avaliar com mais detalhe o encadeamento de raciocínios sobre os conceitos e a realização de questões de programação (numa proporção típica de 50/50).*

*Os testes de autoavaliação voluntários permitem aos estudantes avaliar o estado dos seus conhecimentos e servem para despoletar dúvidas e revisões de assuntos específicos nas aulas teórico-práticas. Também constituem uma oportunidade para os estudantes refletirem sobre o processo de ensino/aprendizagem.*

*A metodologia de avaliação distribuída foi escolhida por se adaptar melhor aos hábitos de estudo dos estudantes chegados do ensino secundário. Contudo, vários estudos internos mostram que o insucesso no primeiro ano pode ter um impacto desproporcional no percurso académico dos estudantes. Daí serem existentes os testes de repescagem, que dão uma possibilidade adicional a estudantes previamente sem aprovação de demonstrarem que atingiram um nível mínimo aceitável de domínio dos temas da unidade curricular.*

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

*The first test evaluates the syllabus contents of the modules M1-M4 (essentially digital systems) and the second test the contents of modules M5-M10 modules (processor and memory). It is a natural division of the contents, which are thus completely covered by the evaluation.*

*The tests include both multiple choice and open-ended questions (in a typical 50/50 ratio); the latter allow the evaluation in more detail of the chain of reasoning about syllabus concepts used by the students and the inclusion of programming problems.*

*Voluntary self-assessment tests allow students to assess the state of their knowledge and serve to trigger questions and revisions of specific subjects in the theoretical-practical sessions. They also provide an opportunity for students to reflect on the teaching / learning process.*

*The distributed evaluation methodology was chosen because it best adapts to the study habits of students arriving from secondary education. However, several internal studies show that failure in the first year can have a disproportionate impact on students' academic path. Hence, there are recap tests, which give an additional possibility to previously unapproved students to demonstrate that they have achieved an acceptable minimum level of mastery of the course subjects.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Patterson, D., & Hennessy, J. (2016). Computer Organization and Design: The Hardware/Software Interface ARM Edition. Elsevier.*

*Harris, S., & Harris, D. (2016). Digital Design and Computer Architecture ARM Edition. Elsevier.*

*Baptista, C. P. (2015). Introdução aos Sistemas Digitais. FCA.*

### **Mapa IV - Inteligência Artificial**

#### **4.4.1.1. Designação da unidade curricular:**

*Inteligência Artificial*

#### **4.4.1.1. Title of curricular unit:**

*Artificial Intelligence*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

*52 (26T + 26TP)*

#### **4.4.1.6. ECTS:**

*6*

#### **4.4.1.7. Observações:**

*Esta unidade curricular apresenta um conjunto de assuntos nucleares para a área da Inteligência Artificial (IA) e dos Sistemas Inteligentes.*

#### 4.4.1.7. Observations:

*This course provides a set of subjects (topics) that are the core of the Artificial Intelligence and Intelligent System area.*

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

*Luís Paulo Gonçalves dos Reis (T - 26h - 2 turmas; TP – 26h – 11 turmas)*

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

*Inês Castro Dutra (T - 26h - 2 turmas; TP - 26h - 11 turmas)*

*Alípio Mário Guedes Jorge (TP - 52h - 11 turmas)*

*Ana Paula Cunha da Rocha (TP - 52h - 11 turmas)*

*Henrique Daniel de Avelar Lopes Cardoso (TP - 52h - 11 turmas)*

*Docente a contratar (TP - 78h - 11 turmas)*

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

*Os objetivos principais são:*

- 1. Compreender os fundamentos da Inteligência Artificial e dos Sistemas Inteligentes, o que os caracteriza e distingue e qual a sua aplicabilidade.*
- 2. Ser capaz de projetar e implementar Agentes e Sistemas Multi-Agente para resolver diferentes problemas.*
- 3. Aprender métodos e algoritmos heurísticos e sistemáticos de resolução de problemas, com e sem adversários.*
- 4. Aprender métodos de aquisição, representação e manipulação do Conhecimento impreciso utilizando diferentes formalismos.*
- 5. Compreender as bases do processamento da linguagem natural e suas aplicações.*
- 6. Conhecer e ser capaz de aplicar algoritmos de aprendizagem com diferentes paradigmas (supervisionada, evolucionária, por reforço, em profundidade).*
- 7. Conhecer tópicos avançados em IA e ser capaz de formular uma visão sobre o futuro da IA.*
- 8. Desenvolver projetos simples, mas completos, usando técnicas de IA.*

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

*The main objectives are:*

- 1. Understand the fundamentals of Artificial Intelligence and Intelligent Systems, what characterizes and distinguishes them and their applicability.*
- 2. Being able to design and implement Agents and Multi-Agent Systems to solve different problems.*
- 3. To learn heuristic and systematic methods of problem solving, with and without adversaries.*
- 4. To learn methods of acquisition, representation and reasoning with uncertain knowledge using different formalisms.*
- 5. To understand the basis of natural language processing and its applications.*
- 6. To know and be able to use machine-learning algorithms applying different paradigms (supervised, evolutionary, reinforcement, deep learning).*
- 7. To understand advanced topics in Artificial Intelligence and be able to formulate a vision into the future of AI.*
- 8. To develop simple but complete projects using AI techniques*

#### 4.4.5. Conteúdos programáticos:

*1. Introdução à Inteligência Artificial (IA)*

*Definição de IA; Fundamentos, Âmbito, Evolução e Cronologia da IA; Aplicações da IA.*

*2. Agentes Inteligentes*

*Agente; Ambientes; Arquiteturas de Agentes; Sistemas Multi-Agente.*

*3. Métodos de Resolução de Problemas*

*Formulação de Problemas; Espaço de Estados; Estratégias de Pesquisa; Pesquisa com Adversários e Minimax.*

*4.. Otimização e Meta-heurísticas*

*Formulação de Problemas de Decisão/Otimização. Metaheurísticas; Algoritmos Genéticos; Satisfação de Restrições.*

*5. Engenharia do Conhecimento*

*Aquisição, Representação e Manipulação do Conhecimento e Raciocínio.*

*6. Aprendizagem Computacional*

*Aprendizagem Não Supervisionada; Supervisionada; Redes Neurais e SVMs; Por Reforço; Profunda.*

*7. Processamento de Linguagem Natural*

*Aproximação Clássica; Abordagem Estatística; Mineração de Texto.*

*8. Tópicos Avançados de IA*

*O Futuro da IA. IA e a Sociedade. IA Benéfica. IA Explicável; Máquinas Éticas. IA fraca/forte. Super Inteligência. Singularidade.*

#### 4.4.5. Syllabus:

*1. Introduction to Artificial Intelligence (AI)*

*Definition of AI; Fundamentals, Scope, Evolution and Chronology of AI; AI Applications.*

*2. Intelligent Agents*

*Concept of Agent; Environments; Agent Architectures; Multi-Agent Systems.*

*3. Problem Solving Methods*

*Problem Formulation. State Space. Search Strategies and A\*; Search with Adversaries and Minimax.*

**4. Optimization and Metaheuristics**

*Formulation of Decision/Optimization Problems; Metaheuristics; Genetic Algorithms; Constraint Satisfaction.*

**5. Knowledge Engineering**

*Knowledge Acquisition, Representation and Reasoning.*

**6. Machine Learning**

*Unsupervised Learning; Supervised Learning; Neural Networks and Support Vector Machines; Reinforcement Learning; Deep Learning.*

**7. Natural Language Processing**

*Classical Approach; Statistical Approach; Text Mining.*

**8. Advanced Topics in Artificial Intelligence**

*The Future of AI. IA and the Society. Beneficial IA. Explainable AI, Machine Ethics. Weak and Strong IA. Super Intelligence. The Singularity.*

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

*Os conteúdos programáticos exploram os principais conceitos da Inteligência Artificial, começando com uma introdução ao tema, seguida pelo conceito de Agente Inteligente, pela formulação e resolução de problemas utilizando pesquisa com e sem adversários e pela formulação e resolução de problemas de otimização utilizando meta-heurísticas e algoritmos genéticos. Estes conteúdos permitem cobrir com detalhe os objetivos de aprendizagem 1 a 3. Os conteúdos programáticos seguintes exploram a Engenharia do Conhecimento, a Aprendizagem Computacional, o Processamento de Linguagem Natural e alguns tópicos avançados de IA com ênfase para a IA no futuro, o que permite cobrir os objetivos de aprendizagem 4 a 7. A realização de dois trabalhos práticos, o primeiro focado na primeira metade da matéria e o segundo na parte final da matéria permite cobrir o objetivo de aprendizagem final.*

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

*The syllabus explores the main concepts of Artificial Intelligence, starting with an introduction to the theme, followed by the concept of Intelligent Agent, problem formulation and resolution using search with and without opponents, and optimization problem formulation and resolution using metaheuristics and genetic algorithms. These contents cover the learning objectives 1 to 3 in detail. The rest of the syllabus explores Knowledge Engineering, Machine Learning, Natural Language Processing and some Advanced AI topics with an emphasis on AI in the future, covering learning objectives 4 to 7. Two practical assignments, the first focusing on the first half of the contents and the second on the final part of the contents, cover the final learning objective.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*Exposição com interação nas Aulas teóricas. Exercícios de modelação e resolução de problemas, de programação e desenvolvimento de projeto nas Aulas teórico-práticas.*

*Tipo de Avaliação: Avaliação distribuída sem exame final*

*Condições de Frequência: Não exceder o número limite de faltas e obter nota  $\geq 7.5$  (em 20) na componente de Avaliação Distribuída (AD)*

*Fórmula de avaliação: Componente de Avaliação Distribuída (AD): peso=50%, inclui dois trabalhos práticos:*

- Trabalho1 (qualidade do trabalho e código, relatório/artigo e qualidade da demonstração) (40%);
- Trabalho2 (qualidade do Trabalho e código, relatório/artigo e qualidade da demonstração) (40%);
- Avaliação durante as aulas: minitests/atividades moodle (20%)

*Componente de Testes/Exames (CE): peso=50% (prova com consulta, com a duração de 2h30m).*

*A aprovação implica obtenção de nota  $\geq 7.5$  (em 20) em cada uma das duas componentes de avaliação, classificação distribuída (CD) e exame (CE).*

**4.4.7. Teaching methodologies (including students' assessment):**

*Theoretical classes: exposition with interaction. Theoretical-practical classes: modeling, problem-solving, programming exercises and project development.*

*Type of evaluation: Distributed evaluation without final exam*

*Terms of frequency: Not exceed the absence limit allowed and have a minimum of 37,5% in the distributed classification (DC)*

*Formula of evaluation: Distributed Classification (DC): weight=50%, including two practical works (% related to the distributed classification):*

- Assignment1 (quality of the work and code, final report/article, presentation performance) (40%);
- Assignment2 (quality of the work and code, final report/article, presentation performance) (40%);
- Evaluation during classes: mini-exams/moodle activities (20%)

*Test/Exam (EC): weight=50% (2h30m test with consultation).*

*To pass, the student must have a minimum of 37.5% in each of the two evaluation components, distributed (DC) and exam (EC).*

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

*As aulas teóricas são dedicadas essencialmente à apresentação dos conceitos fundamentais da Inteligência Artificial, realizada recorrendo a exemplos apropriados, incluindo diversos projetos de investigação coordenados pelos docentes da disciplina e que exploram em detalhe os conceitos abordados. Nas aulas teóricas são também realizados exercícios simples e inquéritos (quiz) rápidos online para aferir o grau de compreensão na matéria e permitir uma avaliação formativa e sumativa adequada. Os exemplos e exercícios realizados são disponibilizados na página web da UC para auxiliar o estudo dos estudantes.*

*As aulas teórico-práticas são essencialmente destinadas à resolução de exercícios sobre cada um dos tópicos abordados nas aulas teóricas. O objetivo principal destes exercícios é a aprendizagem dos conceitos básicos da IA, à medida que vão sendo apresentados nas aulas teóricas. Nas aulas práticas é também realizado o seguimento dos dois*

*trabalhos práticos de grupo, sendo fornecido feedback constante aos estudantes sobre o seu trabalho prático. Os trabalhos de grupo, cujo desenvolvimento é realizado essencialmente fora das aulas, por grupos de três estudantes, fomentam a capacidade de trabalho em equipa. A evolução dos projetos é avaliada periodicamente, sendo discutidas as soluções propostas e sugeridos caminhos alternativos, sendo, no entanto, sempre fomentada a criatividade dos estudantes. No final de cada um dos projetos, cada grupo faz uma demonstração do trabalho, respondendo a questões sobre a solução desenvolvida.*

*O regime de avaliação foi concebido para aferir os conhecimentos e competências adquiridas individualmente por cada estudante. A avaliação contém uma componente forte de avaliação de trabalho de grupo (50% da nota final). No entanto, embora os dois trabalhos sejam realizados em grupo, são avaliados individualmente nas sessões de demonstração e apresentação. A avaliação contém também uma componente exclusivamente individual (exame e participação nas aulas de 50%).*

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

*The lectures are devoted essentially to the presentation of the fundamental concepts of Artificial Intelligence, using appropriate examples, including several research projects coordinated by the teachers of the discipline and that explore in detail the concepts covered. In the lectures, simple exercises and quick online quizzes are also conducted to measure the degree of understanding in the subject and to enable proper formative and summative assessment. The examples and exercises performed are made available on the UC website to assist student study.*

*Theoretical-practical classes are essentially aimed at solving exercises on each of the topics covered in the theoretical classes. The main purpose of these exercises is to learn the basic concepts of AI as they are presented in the theoretical lectures. In practical classes, the two practical group projects are also followed up and students are given constant feedback on their practical work.*

*Group work, developed essentially out of the classes, by groups of three students, fosters teamwork. The evolution of the projects is periodically evaluated, and the proposed solutions are discussed and alternative paths are suggested, but the students' creativity is always fostered. At the end of each project, each group demonstrates the work, answering questions about the solution developed.*

*The assessment scheme is designed to measure the knowledge and skills acquired individually by each student. The evaluation contains a strong group work evaluation component (50% of the final grade). However, although the two works are done in groups, they are evaluated individually in the demonstration and presentation sessions. The assessment also contains an exclusively individual component (50% exam and class participation).*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Russell, S. & Norvig, P. (2014). Artificial Intelligence: A Modern Approach. 3rd Edition. Pearson Education Limited (4th edition to be released Feb 2020). ISBN: 978-9-332-54351-5.*

*Weiss, G. (editor) (2013). Multiagent Systems, Second Edition. The MIT Press. Intelligent Robotics and Autonomous Agents series, Cambridge, Massachusetts, London England, ISBN: 978-0-262-01889-0*

*Domingos, P. (2015). The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. Basic Books. ISBN: 978-0465065707*

*Costa, E. & Simões, A. (2008). Inteligência Artificial: Fundamentos e Aplicações. Second Edition, FCA, ISBN: 978-9-727-22340-4*

### **Mapa IV - Interação Pessoa Computador**

#### **4.4.1.1. Designação da unidade curricular:**

*Interação Pessoa Computador*

#### **4.4.1.1. Title of curricular unit:**

*Human Computer Interaction*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*121.5*

#### **4.4.1.5. Horas de contacto:**

*39 (19.5T + 19.5TP)*

#### **4.4.1.6. ECTS:**

*4.5*

#### **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):***Rui Pedro Amaral Rodrigues (T - 19.5h - 2 turmas; TP - 19.5h - 10 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Álvaro Pedro de Barros Borges Reis Figueira (T - 19.5h - 2 turmas; TP - 19.5h - 10 turmas)**José Luís Cabral Moura Borges (TP - 19.5h - 10 turmas)**Luis Filipe Pinto de Almeida Teixeira (TP - 19.5h - 10 turmas)**Maria Teresa Galvão Dias (TP - 78h - 10 turmas)**Miguel Tavares Coimbra (TP - 39h - 10 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Objetivos: Contacto com os tópicos seguintes, e sua aplicação prática, no contexto de sistemas interativos:**1. Fundações IPC: Principais conceitos de IPC, perceção humana, processos de cognição, sistemas de interação, e história de IPC**2. Interação com o Utilizador (UI) e Design para a Experiência (UX): metodologias de design de interação, incluindo Design Centrado no Utilizador (UCD) e testes**3. Estudos de utilizadores: avaliar projetos de UI e UX e sua usabilidade**4. Interação multimodal: aplicar conhecimentos na criação de interfaces multimodais com tecnologias interativas**Os resultados de aprendizagem (RA) devem traduzir-se na capacidade de:**1. Analisar sistemas existentes e a qualidade das suas interações e experiências**2. Aplicar metodologias como UCD para estudar desafios de interação e propor, avaliar e aperfeiçoar soluções**3. Utilizar ferramentas para criar e testar protótipos de UI/UX a vários níveis**4. Conduzir estudos com utilizadores, no contexto de UI/UX e usabilidade***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***Main goals: to get contact, practice and experience with the following topics:**1. HCI foundations: Main concepts of HCI, human perception and cognition processes, interactive systems, and abridged history of HCI.**2. User Interaction (UI) and Experience design (UX): methodologies for designing interaction, namely User-Centered Design (UCD) and testing**3. User studies: Assessment of UI and UX designs and their usability**4. Multimodal Interaction: apply the presented topics for creating multimodal interfaces using interactive technologies**The learning outcomes (LO) should be reflected in the ability to:**1. Analyze existing systems and the quality of the interactions and experiences provided**2. Apply UCD and other methodologies to study interaction challenges and propose, evaluate and refine solutions**3. Use appropriate tools for creating and testing UI/UX prototypes at various levels of complexity**4. Define and carry out user studies in the context of UI/UX and usability evaluations***4.4.5. Conteúdos programáticos:***1) Fundamentos de IHC**a) Importância da IHC**b) Conceitos-base**i) Perceção humana e processos cognitivos**ii) Sistemas computacionais e seus mecanismos de interação,**iii) Metodologias e padrões de UI/UX design**2) Design Centrado no Utilizador**a) Identificação de necessidades e definição de requisitos**b) Ideação e desenho**c) Prototipagem**d) Avaliação**e) Iteração e refinamentos**3) UI/UX aplicado**a) Ferramentas para UI/UX Design**b) Interação Multimodal**c) Desenho para dispositivos e ambientes complexos (dispositivos móveis, Realidade Virtual, ...)**d) Integração no processo de criação e desenvolvimento de produto***4.4.5. Syllabus:***1) HCI Foundations**a) Importance of HCI**b) Basis concepts of HCI**i) Human perception and cognition processes,**ii) Computer systems and their interaction mechanisms,**iii) Overview of UI/UX design methodologies and patterns*

**2) User-Centered Design**

- a) Finding needs and defining requirements
- b) Ideation and designing alternatives
- c) Prototyping
- d) Evaluation
- e) Iteration and refinement

**3) Applied UI/UX design**

- a) UI/UX Tools
- b) Multimodal interaction
- c) Designing for complex devices and environments (mobile, VR, ...)
- d) Integration in product creation and development process

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

*A primeira componente do programa – Fundamentos de IHC – fornecerá as bases para a área e contribuirá para a capacidade dos estudantes analisarem criticamente projetos e sistemas quanto à UI/UX (RA1 – Resultado de Aprendizagem 1).*

*As restantes componente – Design Centrado no Utilizador e UI/UX aplicado – cobrirão as metodologias e ferramentas necessárias para levar a cabo processos de desenho, prototipagem e avaliação de UI/UX, e serão reforçadas através da sua aplicação prática nos trabalhos desenvolvidos ao longo do semestre, contribuindo para os resultados de aprendizagem RA2, RA3 e RA4.*

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

*The first component of the program – HCI foundations - will provide the basis for the area and contribute to students' ability to critically analyze designs and systems regarding UI / UX (LO1 - Learning Outcome 1).*

*The remaining components – User-Centered Design and Applied UI / UX Design - will cover the methodologies and tools needed to carry out UI / UX design, prototyping and evaluation processes, and will be reinforced through their practical application in the work developed throughout the course. contributing to learning outcomes LO2, LO3 and LO4.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*Na componente teórica, serão veiculados os principais conceitos, teorias e metodologias da IHC, ilustrados com exemplos, palestras e discussão entre docentes e estudantes. A avaliação individual será através de um miniteste. Na componente prática, os alunos irão trabalhar em pequenos grupos num projeto de UI/UX desenvolvido ao longo do semestre.*

*O tópico será selecionado pelos estudantes com a aprovação dos professores, e pode ser relacionado com projetos de outras unidades curriculares. O trabalho será dividido em três fases; no final de cada fase, cada grupo de trabalho deve produzir e apresentar um relatório sobre o trabalho desenvolvido nessa fase.*

*Tipo de avaliação: Avaliação distribuída sem exame final*

*Fórmula de avaliação:*

$$FG = PG * 80\% + MT * 15\% + CE * 5\%.$$

*\* FG - Nota final*

*\* PG - Nota do projeto (ver abaixo)*

*\* MT - Mini-Teste*

*\* CE - Avaliação Contínua*

$$PG = F1 * 30\% + F2 * 30\% + F3 * 30\% + FP * 10\%.$$

*\* F1, F2, F3 - Fases do Projeto*

*\* FP - Apresentação Final*

**4.4.7. Teaching methodologies (including students' assessment):**

*In the theoretical component, the main concepts, theories and methodologies of HCI will be conveyed, illustrated with examples, case studies, guest lectures and discussions between lecturers and students. This component's individual assessment takes the form of a mini-test.*

*In the practical component, the students will work in groups in a UI/UX project throughout the semester.*

*The topic will be selected by the students with approval of the teachers, and may be related to projects in other courses. The work will be divided in three phases, and at the end of each phase, a report on the work developed in that phase has to be delivered and presented.*

*Type of evaluation: Distributed evaluation without final exam*

*Formula of evaluation:*

$$FG = PG * 80\% + MT * 15\% + CE * 5\%$$

*\* FG - Final grade*

*\* PG - Project grade*

*\* MT - Mini-Test*

*\* CE - Classes' Evaluation*

*Project Grade (PG) is calculated as follows:*

$$PG = F1*30\% + F2*30\% + F3*30\% + FP*10\%$$

*\*F1, F2, F3 - Project Phases*

*\*FP - Final Presentation*

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

*A componente teórica proposta cobre uma parte expositiva e de discussão dos conceitos e metodologias propostas nos quatro objetivos da UC, nomeadamente Fundações IPC, Interação com o Utilizador (UI) e Design para a Experiência (UX), Estudos de utilizadores e Interação Multimodal.*

*A componente prática - o projeto a desenvolver pelos estudantes, e acompanhado pelos docentes - incluirá: selecionar um problema/tópico/aplicação para o qual será projetada uma interface com o utilizador; identificar objetivos e requisitos; utilizar metodologias apropriadas, como definição de personas, histórias/conhecimento de utilizadores, cenários de uso e outros; propor uma UI, criar protótipos de baixa fidelidade; avaliar com utilizadores; iterar sobre o design; reavaliar quando possível. O tópico será selecionado pelos estudantes com a aprovação dos professores, e pode ser relacionado com projetos de outras unidades curriculares, por exemplo, projetando a UI para um programa ou site a ser desenvolvido em outra UC laboratorial. Os projetos serão acompanhados de perto pelos docentes, apoiando a sua evolução e adaptando esse apoio às necessidades específicas de cada projeto.*

*Desta forma, os objetivos O3 e O4 são cobertos e explorados através da aplicação prática dos conceitos. Ao serem expostos a estas práticas, os estudantes ganharão experiência na análise, desenho, prototipagem e avaliação de sistemas de interação em consonância com os resultados de aprendizagem enunciados.*

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

*The proposed theoretical component covers an lecturing and discussion part of the concepts and methodologies proposed in the four objectives of the curricular unit, namely HCI foundations, User Interaction (UI) and Experience design (UX), User studies and Multimodal Interaction.*

*The practical component - the project to be developed by the students, and accompanied by the teachers - will include: selecting a problem/topic/application for which an interface with the user will be designed; identifying objectives and requirements; using appropriate methodologies, such as defining personas, user stories / knowledge, usage scenarios and others; propose a UI, create low-fidelity prototypes; evaluate with users; iterate over the design; reevaluate when possible.*

*The topic will be selected by the students with the approval of the teachers, and can be related to projects from other curricular units, for example, designing the UI for a program or website to be developed in another curricular unit. The projects will be closely monitored by the teachers, supporting their evolution and adapting this support to the specific needs of each project.*

*In this way, objectives 3 and 4 are covered and explored through the practical application of the concepts. When exposed to these practices, students will gain experience in the analysis, design, prototyping and evaluation of interaction systems in line with the stated learning outcomes.*

**4.4.9. Bibliografia de consulta/existência obrigatória:**

*Tufte, E. R. (2001). The visual display of quantitative information (2nd ed). Graphics Press.*  
*Cooper, A., Reimann, R., Cronin, D., & Cooper, A. (2014). About face: The essentials of interaction design (Fourth edition). John Wiley and Sons.*  
*Tidwell, J. (2011). Designing interfaces: Patterns for effective interaction design (2. ed). O'Reilly.*  
*Rogers, Y., Sharp, H., & Preece, J. (2011). Interaction design: Beyond human-computer interaction (3rd ed). Wiley.*  
*Benyon, D. (2013). Designing interactive systems: A comprehensive guide to HCI and interaction design (Third edition). Pearson.*  
*Norman, D. A. (1999). The invisible computer: Why good products can fail, the personal computer is so complex, and information appliances are the solution (1. MIT Press paperback ed).*  
*Fonseca, M. J., Campos, P., Gonçalves, D. (2012). Introdução ao Design de Interfaces. FCA.*  
*Dix, A. (Ed.). (2004). Human-computer interaction (3rd ed). Pearson/Prentice-Hall.*

**Mapa IV - Laboratório de Bases de Dados e Aplicações Web****4.4.1.1. Designação da unidade curricular:**

*Laboratório de Bases de Dados e Aplicações Web*

**4.4.1.1. Title of curricular unit:**

*Database and Web Applications Laboratory*

**4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

**4.4.1.3. Duração:**

*Semestral*

**4.4.1.4. Horas de trabalho:**

*162*

**4.4.1.5. Horas de contacto:**

52 (26T + 26PL)

**4.4.1.6. ECTS:**

6

**4.4.1.7. Observações:**

&lt;sem resposta&gt;

**4.4.1.7. Observations:**

&lt;no answer&gt;

**4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***Sérgio Sobral Nunes (T - 26h - 2 turmas; PL - 26h - 10 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***José Paulo de Vilhena Geraldês Leal (T - 26h - 2 turmas; PL - 26h - 10 turmas)**Rolando da Silva Martins (PL - 52h - 10 turmas)**Tiago Boldt Sousa (PL - 78h - 10 turmas)**Docente a contratar (PL - 78h -10 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):**

*LBAW é uma unidade curricular laboratorial que tem como objetivo consolidar e desenvolver tópicos da área de Bases de Dados e de Linguagens e Tecnologias Web expostas em unidades curriculares anteriores. Oferece uma perspetiva prática sobre estas duas áreas através do desenvolvimento estruturado de um sistema de informação baseado na web.*

*Ao completar esta unidade curricular, o estudante deve ser capaz de projetar, documentar e implementar um sistema de informação baseado em tecnologias web e suportado por um sistema de gestão de base de dados. Em particular, o estudante deve ser capaz de:*

- 1. Especificar os requisitos do sistema de informação;*
- 2. Especificar e implementar a camada de dados do sistema de informação;*
- 3. Desenhar e implementar mecanismos para manter a integridade e otimizar o acesso aos dados;*
- 4. Especificar e implementar a camada web do sistema de informação;*
- 5. Desenhar e implementar aplicações web para dispositivos móveis com recurso a tecnologias web.*

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

*LBAW is a laboratory course that aims to consolidate and develop topics from the areas of Databases and Web Languages and Technologies exposed in previous courses. It offers a practical perspective on these two areas through the structured development of a web-based information system.*

*Upon completing this course, the student must be able to design, document and implement an information system based on web technologies and supported by a database management system. In particular, the student must be able to:*

- 1. Specify the information system requirements;*
- 2. Specify and implement the data layer of the information system;*
- 3. Design and implement mechanisms to maintain integrity and optimize access to data;*
- 4. Specify and implement the web layer of the information system;*
- 5. Design and implement web applications for mobile devices using web technologies.*

**4.4.5. Conteúdos programáticos:**

- 1. Desenvolvimento de uma aplicação web suportada por uma base de dados: levantamento de requisitos, conceção, modelação, documentação e implementação.*
- 2. Identificação e documentação de requisitos com recurso a metodologias ágeis.*
- 3. Desenho e implementação de uma base de dados relacional num paradigma cliente-servidor: modelação de dados com recurso a UML; obtenção e normalização do esquema relacional; implementação de acessos com recurso a SQL; otimização dos acessos com recurso a vistas, índices de desempenho, e índices para pesquisa de texto integral; manutenção da integridade dos dados com gatilhos, transações e procedimentos.*
- 4. Desenho e implementação de uma aplicação web: desenho da arquitetura web; implementação com recurso a tecnologias web baseadas no servidor e no cliente; utilização de frameworks e bibliotecas web no desenvolvimento; implementação de uma interface para dispositivos móveis com recurso a tecnologias web.*
- 5. Noções gerais de usabilidade e acessibilidade web.*

**4.4.5. Syllabus:**

1. *Development of a web application supported by a database: requirements gathering, design, modeling, documentation and implementation.*
2. *Identification and documentation of requirements using agile methodologies.*
3. *Design and implementation of a relational database in a client-server paradigm: data modeling using UML; obtaining and normalizing the relational schema; implementation of accesses using SQL; optimization of accesses using views, performance indexes, and indexes for full text search; maintaining data integrity with triggers, transactions and procedures.*
4. *Design and implementation of a web application: design of the web architecture; implementation using web technologies based on the server and the client; use of web frameworks and libraries in development; implementation of an interface for mobile devices using web technologies.*
5. *General notions of usability and web accessibility.*

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

*Esta é uma unidade laboratorial que tem como objetivo consolidar conhecimentos adquiridos em unidades anteriores tendo em vista a aplicação prática desses conhecimentos na construção de um sistema de informação disponível via web. Os conteúdos programáticos estão organizados de forma a enquadrar as diferentes fases de desenvolvimento do projeto, nomeadamente: a análise e especificação de requisitos; o desenho e modelação da base de dados; o desenho e modelação da arquitetura da camada web; e a implementação do produto com recurso a ferramentas para a gestão do código fonte e para a gestão da infraestrutura através de virtualização. São também abordados conceitos gerais de usabilidade e acessibilidade web tendo em vista a aplicação prática destes princípios.*

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

*This is a laboratory course that aims to consolidate knowledge acquired in previous courses with the goal of applying this knowledge in the development of an information system available via the web. The syllabus contents are organized in order to fit the different phases of the project's development, namely: the analysis and specification of requirements; the design and modelling of the database; the design and modelling of the web layer architecture; and the implementation of the product using tools for source code management and infrastructure management through virtualization. General concepts of usability and web accessibility are also addressed with a view to the practical application of these principles.*

#### **4.4.7. Metodologias de ensino (avaliação incluída):**

*As aulas teóricas são usadas para a exposição dos conceitos e das tecnologias a usar nos projetos, para a apresentação de guiões relativos aos artefactos, e para a discussão de exemplos. As aulas laboratoriais são usadas para o desenvolvimento do projeto, em grupo, e para o acompanhamento e discussão sobre os artefactos produzidos pelo grupo.*

*Regime de avaliação distribuída sem exame final.*

*O projeto é avaliado através do acompanhamento semanal nas aulas laboratoriais e a entrega dos artefactos previstos. Os conceitos teóricos são avaliados através de uma prova individual com perguntas de escolha múltipla.*

*Fórmula de avaliação: 80% Projeto + 20% Prova Individual*

*Projeto = 20% ER+ 20% EBD + 20% EAP + 40% PA*

*ER-Especificação de Requisitos*

*EBD-Especificação da Base de Dados*

*EAP-Especificação da Arquitetura e Protótipo Vertical*

*PA-Produto e Apresentação*

*A aprovação na unidade curricular está condicionada à obtenção de 40% na prova individual e de 50% em cada um dos componentes do projeto.*

#### **4.4.7. Teaching methodologies (including students' assessment):**

*Theoretical classes are used to expose the concepts and technologies to be used in the projects, to present scripts related to the artefacts, and to discuss examples. Laboratory classes are used for the development of the project, in groups, and for the monitoring and discussion of the artifacts produced by each group.*

*Evaluation model distributed without final exam.*

*The project is evaluated through weekly monitoring in laboratory classes and the delivery of planned artifacts.*

*Theoretical concepts are evaluated through an individual test with multiple choice questions.*

*Evaluation formula: 80% Project + 20% Individual Test*

*Project = 20% ER + 20% EBD + 20% EAP + 40% PA*

*ER - Requirements Specification*

*EBD - Database Specification*

*EAP - Architecture Specification and Vertical Prototype*

*PA - Product and Presentation*

*The approval in the curricular unit is conditional on obtaining 40% in the individual test and 50% in each of the components of the project.*

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

*Considerando tratar-se de uma unidade curricular laboratorial, baseada no desenvolvimento de um projeto de média dimensão ao longo do semestre, os temas tratados nas aulas acompanham as fases de desenvolvimento do projeto e estão organizadas em quatro grandes grupos: especificação de requisitos; especificação e implementação da base de dados; especificação e implementação de um protótipo da arquitetura web; implementação do produto final. Neste contexto, as aulas teóricas organizam-se com apresentação dos tópicos associados às fases do projeto, incluindo a apresentação e discussão de casos práticos. Durante a fase de implementação final, nas aulas teóricas são explorados tópicos de ligação à indústria na área do desenvolvimento de aplicações web. As aulas laboratoriais são*

*focadas no desenvolvimento, apresentação e avaliação dos artefactos que compõem cada uma das fases de desenvolvimento do projeto (especificação, base de dados, arquitetura web, implementação).*

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

*Considering that it is a laboratory curricular unit, based on the development of a medium-sized project throughout the semester, the themes covered in the classes follow the project's development phases and are organized in four large groups: specification of requirements; specification and implementation of the database; specification and implementation of a prototype of the web architecture; implementation of the final product. In this context, the theoretical classes are organized with the presentation of topics associated with the project phases, including the presentation and discussion of practical cases. During the final implementation phase, the theoretical lessons explore topics related to the industry in the area of web application development. Laboratory classes are focused on the development, presentation and evaluation of the artefacts that make up each of the project's development phases (specification, database, web architecture, implementation).*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Ramakrishnan, R., & Gehrke, J. (2018). Database management systems. McGraw-Hill.  
Zalewski, M. (2012). The tangled Web: A guide to securing modern web applications. No Starch Press.  
Ullman, J. D., & Widom, J. (2007). A first course in database systems. Prentice Hall.  
Miles, R., & Hamilton, K. (2006). Learning UML 2.0. O'Reilly.  
Nielsen, J. (1999). Designing web usability: The practice of simplicity. New Riders Publishing.*

### **Mapa IV - Laboratório de Computadores**

#### **4.4.1.1. Designação da unidade curricular:**

*Laboratório de Computadores*

#### **4.4.1.1. Title of curricular unit:**

*Computer Laboratory*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

*52 (26T + 26PL)*

#### **4.4.1.6. ECTS:**

*6*

#### **4.4.1.7. Observações:**

*<sem resposta>*

#### **4.4.1.7. Observations:**

*<no answer>*

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

*Pedro Miguel Alves Brandão (T - 26h - 2 turmas; PL - 52h - 12 turmas)*

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

*Pedro Alexandre Guimarães Lobo Ferreira do Souto (T - 26h - 2 turmas; PL - 52h - 12 turmas)  
José Manuel de Magalhães Cruz (PL - 26h - 12 turmas)  
Pedro Miguel Moreira da Silva (PL - 78h - 12 turmas)  
Docentes a contratar (PL - 104h - 12 turmas)*

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

*Os objetivos da unidade curricular são dotar os estudantes com conhecimentos essenciais e experiência para serem capazes de:  
- usar a interface de "hardware" de dispositivos de entrada/saída;;*

- desenvolver "software" de baixo nível, p.ex. "device drivers", e de "software" embebido
- usar a linguagem de programação C de modo estruturado;
- desenvolver programas codificados em C e em "assembly"
- fazer "debugging" de forma sistemática (com base no método experimental científico)
- usar várias ferramentas de desenvolvimento de software (compiladores e compiladores cruzados, "assemblers", "linkers", build automation tools, version control systems)

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

*This course unit aims to endow students with the knowledge and skills required to:*

- use the hardware interface offered by I/O devices
- develop low level software, e.g. "device drivers", and embedded software;
- program in the C language (using C++ as a reference);
- develop programs combining C and "assembly"
- apply systematic debugging techniques (i.e. based on the experimental scientific method)
- use various software development tools (compilers and cross-compilers, assemblers, linkers, build automation tools, version control systems)

#### 4.4.5. Conteúdos programáticos:

*Periféricos de entrada/saída e seus modos de funcionamento. Device drivers. Mapeamento direto no espaço de endereçamento de memória. Acesso a periféricos no modo "polled" e por interrupção. Interrupções nos processadores IA-32 e ARM, e respetivos controladores de interrupção. Escrita de rotinas de interrupção em "assembly" e em C. Periféricos típicos de um PC e de plataformas embebidas.*

*Programação na linguagem C: principais diferenças em relação à linguagem C++, métodos para estruturação do código. Organização da memória de um processo. Funções: mecanismo de chamada, passagem de parâmetros, armazenamento local e retorno de valores. Programação baseada em eventos. Máquinas de estado. Programação combinada em C e "assembly" dos processadores ARM. Técnicas sistemáticas para "debugging" baseadas no método científico experimental.*

*Criação e utilização de bibliotecas. Ligação estática de código objeto.*

*Ferramentas de desenvolvimento de software: cc, make, ar, diff, patch, SVN, doxygen.*

#### 4.4.5. Syllabus:

*Input/Output peripherals and their operation. Device drivers. Direct mapping in a process address space. Access to peripherals in polled mode and by interrupt. Processor interrupts in the IA-32 and ARM architectures and respective interrupt controllers. Interrupt service routines in Assembly and in C. Study of common personal computer and embedded systems I/O devices.*

*Programming in the C programming language: main differences with respect to C++ language; structured programming in C. Memory layout of a process. Function calls: mechanisms, parameter passing, storage of local variables and return values. Event based programming. State machines. Combined programming in C and the IA-32 processor family assembly. Systematic debugging techniques based on the experimental scientific method.*

*Creation and use of libraries. Static linking of object code.*

*Use of software development tools: cc, make, ar, diff, patch, SVN and doxygen.*

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

*A maioria dos conteúdos programáticos estão diretamente relacionados com um ou mais objetivos de aprendizagem, usando inclusive palavras e termos técnicos comuns. Os objetivos de aprendizagem estão relacionados com a programação de dispositivos de E/S, o que exige o estudo da organização destes dispositivos e de como eles interagem com outros componentes de "hardware", e.g. o processador, bem como das técnicas de "hardware" usadas por estes dispositivos, p.ex. interrupções. Além disso, para realizar a interface com dispositivos de E/S são necessárias técnicas de programação específicas, p.ex. máquinas de estado, ditas de "baixo-nível", que são cobertas pelos tópicos enumerados. A sistematização de técnicas de "debugging" baseadas no método experimental científico é essencial para a sua aplicação. Finalmente, a apresentação dos conceitos associados a ferramentas de desenvolvimento e construção de "software", p.ex. make e svn, facilitam a utilização proficiente destas ferramentas.*

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

*The majority of the syllabus topics are directly related to one or more of the learning outcomes: actually, we often use the same words and technical terms in both of them. The learning outcomes are focused on the interfacing with I/O devices, and therefore require the study of the structure of these devices, how they interface with other hardware components, e.g. the processor, as well as the study of hardware techniques used by these devices, e.g. interrupts. Furthermore, interfacing with I/O devices requires the use of specific (low-level) programming techniques, e.g. state machines, that are covered in the listed topics. The awareness of systematic debugging techniques is prerequisite for their application. Likewise, the understanding of the concepts underpinning different software development tools, e.g. make or svn, are instrumental for a more proficient use of these tools.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*"Aprender fazendo". Nas primeiras 10 das aulas laboratoriais os estudantes realizam em grupos de 2 estudantes pequenos trabalhos práticos sobre periféricos comuns do PC ou em plataformas computacionais para sistemas embebidos e aplicando técnicas de programação específicas. Nas restantes aulas laboratoriais será realizado um projeto proposto pelos estudantes que deverá integrar a maioria dos periféricos e técnicas estudadas.*

*Nas aulas teóricas faz-se a exposição dos conceitos, tendo sempre em vista a sua aplicação prática, e discute-se os pormenores necessários à realização dos trabalhos práticos e do projeto.*

*Fórmula de cálculo da nota final:  $0,45 * \text{Trabalhos} + 0,45 * \text{Projeto} + 0,1 * \text{<Prova de Programação>}$  onde "Trabalhos" é a média da classificação de 3 trabalhos à escolha do estudante entre os 5 avaliados.*

*<Prova de Programação> é um teste de programação individual onde os estudantes deverão usar um dispositivo periférico semelhante aos usados nos trabalhos.*

#### **4.4.7. Teaching methodologies (including students' assessment):**

*A "learn by doing" approach is used. In the first 10 lab classes, students carry out lab assignments about some of the most common PC peripherals or the I/O devices of an embedded systems platform. In the remaining weeks, students will develop a small integration project using several peripherals and the techniques previously studied. Students work both on the lab assignments in groups of 2 students.*

*Lecture classes will be used to present the concepts, taking into account their application in the lab assignments, and to discuss the details of the lab assignments and the project.*

*The final grade:  $0,45 * \text{Labs} + 0,45 * \text{Project} + 0,1 * \text{Programming-Test}$  where:*

*"Labs" is the average of the grade of 4 labs chosen by the student.*

*"Programming-Test is an individual programming test in which students must program an I/O device similar to those used in the labs.*

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

*Tratando-se duma unidade curricular laboratorial, os objetivos de aprendizagem são essencialmente de aplicação de conhecimento. Assim, a metodologia usada é centrada num conjunto de trabalhos de programação envolvendo dispositivos de E/S. Para ajudar os estudantes a construir um modelo mental bem estruturado, nas aulas teóricas apresenta-se os conceitos fundamentais ao nível do "hardware" e do "software" e discutem-se pormenores de implementação, que permitam que os estudantes atinjam os objetivos de aprendizagem enunciados mais facilmente. Embora se obrigue os estudantes a usar as diferentes ferramentas de desenvolvimento de programas desde os primeiros trabalhos laboratoriais, o projeto de integração expõe os estudantes a um uso dessas mesmas ferramentas em programas de dimensão comparável a pequenos projetos profissionais. A elevada carga de trabalho de programação dá muitas oportunidades aos estudantes de errarem, e consequentemente de aplicarem técnicas sistemáticas de "debugging".*

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

*As a lab course, the learning outcomes are mainly related to the application of knowledge. Thus the methodology used is centered on a set of programming labs that use different I/O devices. To help the students build a structured mental model, the lectures are used to introduce the main concepts, both of the hardware and the software and to discuss implementation aspects, to help the students meet the stated learning outcomes more easily. Although we require the students to use the different software tools since the first lab assignment, the integration project exposes the students to the use of these tools in programs whose size is comparable to that of small professional projects. The heavy programming load gives students the opportunity to make bugs, and therefore apply systematic debugging techniques.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*White, E. (2011). Making Embedded Systems: Design Patterns for Great Software. O'Riley.*

*Molloy, Derek (2019). Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux (2nd Ed.).Wiley*

### **Mapa IV - Laboratório de Desenho e Teste de Software**

#### **4.4.1.1. Designação da unidade curricular:**

*Laboratório de Desenho e Teste de Software*

#### **4.4.1.1. Title of curricular unit:**

*Software Design and Testing Laboratory*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

52 (26T + 26PL)

**4.4.1.6. ECTS:**

6

**4.4.1.7. Observações:**

&lt;sem resposta&gt;

**4.4.1.7. Observations:**

&lt;no answer&gt;

**4.4.2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo):***André Monteiro de Oliveira Restivo (T - 26h - 2 turmas; PL - 52h - 12 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Bruno Serra Loff Barreto (T - 26h - 2 turmas; PL - 26h - 12 turmas)**Filipe Alexandre Pais de Figueiredo Correia (PL - 52h - 12 turmas)**Nuno Honório Rodrigues Flores (PL - 78h - 12 turmas)**Docentes a contratar (PL - 104h - 12 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Pretende-se, nesta unidade curricular, que os estudantes desenvolvam competências no desenho e teste de aplicações usando o paradigma orientado a objetos.**Pretende-se que os estudantes que obtenham aprovação à unidade curricular, consigam:*

- 1. Usar o Git como ferramenta de controlo de versões;*
- 2. Usar a linguagem Java como linguagem orientada a objetos;*
- 3. Desenhar testes unitários usando JUnit;*
- 4. Aplicar os princípios SOLID no desenho de aplicações e entender a sua importância;*
- 5. Aplicar design patterns para resolver problemas no desenho de aplicações;*
- 6. Identificar code smells em código orientado a objetos;*
- 7. Usar técnicas de refactoring para melhorar atributos não-funcionais e resolver code smells;*
- 8. Aplicar o padrão arquitetural MVC no desenho de aplicações event-driven;*
- 9. Aplicar os conhecimentos adquiridos no desenho, codificação e teste de aplicações para desktop e para dispositivos móveis.*

**4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***This curricular unit is intended for students to develop application design and testing skills using the object-oriented paradigm.**Students who pass the course should be able to:*

- 1. Use Git as a version control system;*
- 2. Use the Java language as an object-oriented language;*
- 3. Design unit tests using JUnit;*
- 4. Apply SOLID principles in the design of applications and understand their importance;*
- 5. Apply design patterns to solve design problems in application development;*
- 6. Identify code smells in object-oriented code;*
- 7. Use refactoring techniques in order to improve non-functional attributes and solve code smells;*
- 8. Apply the MVC architectural pattern in the design of event-driven applications.*
- 9. Apply the knowledge acquired in the design, coding and testing of desktop and mobile applications.*

**4.4.5. Conteúdos programáticos:**

- 1. Introdução ao Git: repositórios locais, remotes, branches e workflows;*
- 2. Gestão de dependências usando Gradle;*
- 3. A linguagem Java: Tipos, literais e variáveis; Estruturas de controlo; Classes e objetos; Coleções; Genéricos; Threads; Input e output; Interfaces gráficas com Swing;*
- 4. Princípios da programação orientada a objetos: abstração de dados, polimorfismo, herança e encapsulamento; polimorfismo por composição ;*
- 5. Princípios SOLID: Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation e Dependency Inversion;*
- 6. Testes unitários com JUnit: Mocks e Stubs usando Mockito; Cobertura e testes de mutação;*
- 7. Diagramas UML: classes, sequência e comunicação;*
- 8. Code smells e técnicas de refactoring;*
- 9. Padrões de desenho: Factory-Method, Command, Composite, Observer, Strategy, Abstract Factory, State, Adapter, Decorator e Singleton;*
- 10. Model-View-Controller como padrão de arquitetura.*
- 11. Aplicação ao desenvolvimento de aplicações para desktop e dispositivos móveis.*

**4.4.5. Syllabus:**

1. *Introduction to Git: local repositories, remotes, branches and workflows;*
2. *Dependency management using Gradle;*
3. *The Java language: Type, literals and variables; Control structures; Classes and objects; Collections; Generic types; Threads; Input and output; Graphical interfaces with Swing;*
4. *Principles of object-oriented programming: data abstraction, polymorphism, inheritance and encapsulation; polymorphism with composition;*
5. *SOLID principles: Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation and Dependency Inversion;*
6. *Unit testing using JUnit: Mocks and Stubs using Mockito; Coverage and mutation-testing;*
7. *UML diagrams: class, sequence and communication;*
8. *Code smells and refactoring techniques;*
9. *Design patterns: Factory-Method, Command, Composite, Observer, Strategy, Abstract Factory, State, Adapter, Decorator and Singleton;*
10. *The Model-View-Controller architectural pattern.*
11. *Application to the development of desktop and mobile applications.*

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

- *OA1: Pretende-se com uma pequena introdução ao Git que os estudantes fiquem habilitados a usar esta ferramenta como sistema de controlo de versões;*
- *OA2: A linguagem Java será extensivamente apresentada, incluindo conceitos avançados como tipos genéricos;*
- *OA3: Uma introdução à utilização de testes unitários, e uma abordagem a alguns tópicos avançados, irão permitir que os usem esta técnica com segurança;*
- *OA4: Os princípios SOLID serão apresentados e discutidos;*
- *OA5: Três tipos de diagramas UML serão apresentados de forma a serem usados como suporte para a descrição design patterns. Vários design patterns serão apresentados e discutidos;*
- *OA6 e OA7: Serão apresentados os vários tipos de code-smells e a forma como estes podem ser tratados usando técnicas de refactoring;*
- *OA8: O padrão arquitetural MVC será apresentado, assim como as vantagens da sua utilização.*
- *OA9: Atingido através de projeto.*

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

- *LO1: A brief introduction to Git is intended for students to be able to use this tool as a version control system;*
- *LO2: The Java language will be extensively introduced, including advanced concepts such as generic types;*
- *LO3: An introduction to the use of unit testing, and an approach to some advanced topics, will enable the students to use this technique safely;*
- *LO4: The SOLID principles will be presented and discussed;*
- *LO5: Three types of UML diagrams will be presented for use as support for the description of design patterns. Several design patterns will be presented and discussed;*
- *LO6 and LO7: The various types of code smells and how they can be treated using refactoring techniques will be presented;*
- *LO8: The MVC architectural standard will be presented as well as the advantages of its use.*
- *LO9: Achieved through a project.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*Aulas:*

1. *Aulas teóricas de exposição e discussão dos tópicos programáticos;*
2. *Aulas práticas em laboratório:*
  - a. *Na primeira parte (7 semanas) os estudantes irão resolver exercícios que cobrem os conteúdos programáticos;*
  - b. *Na segunda parte (6 semanas) os estudantes irão desenvolver um pequeno projeto integrador.*

*Condições para obtenção de frequência: Não exceder o número limite de faltas e participar ativamente na elaboração do projeto.*

*Avaliação: distribuída sem exame final.*

*Fórmula de avaliação:*

1. *5% - participação nas aulas práticas;*
2. *20% - prova prática individual de programação em computador;*
3. *55% - realização do projeto integrado (grupos de 2 elementos);*
4. *20% - prova individual de escolha múltipla em computador.*

*Todas as componentes têm uma classificação mínima de 40%.*

*A classificação do projeto integrado é composta por:*

1. *10% - relatório intermédio*
2. *30% - relatório final*
3. *60% - produto desenvolvido*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Lessons:*

1. *Theoretical classes for the recitation and discussion of the program topics;*

**2. Practical classes in the laboratory:**

- a. In the first part (7 weeks) students will solve exercises that cover the course syllabus;
- b. In the second part (6 weeks) students will develop a small integrative project.

**Conditions for obtaining frequency:** Do not exceed the limit number of absences and actively participate in the preparation of the project.

**Assessment:** distributed without final exam.

**Evaluation Formula:**

- 5% - participation in practical classes;
- 20% - individual computer programming practical test;
- 55% - completion of the integration project (groups of 2 elements);
- 20% - individual multiple choice computer test.

All components have a minimum grade of 40%.

**The integrated project classification consists of:**

- 10% - intermediate report
- 30% - final report
- 60% - project code

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

- Todos conteúdos programáticos serão expostos durante as aulas teóricas e posteriormente discutidos usando exemplos.
- Durante a primeira parte do semestre, os estudantes irão resolver exercícios práticos que irão cobrir os objetivos de aprendizagem de OA1 a OA7. Estes exercícios são apresentados na forma de tutoriais em que os estudantes são impelidos, não só a resolver problemas, mas também a perceber a razão pela qual esses exercícios são resolvidos da forma sugerida.
- Na segunda parte do semestre, os estudantes irão desenvolver um projeto integrador que irá exercitar novamente os objetivos de aprendizagem de OA1 a OA7, mas também o OA8.

**Avaliação:**

- A avaliação da participação nas aulas práticas servirá para que os estudantes se empenhem na realização dos exercícios de forma a que estes cumpram a sua função pedagógica;
- A prova prática individual em computador irá avaliar os objetivos de aprendizagem OA2 e parte do OA3;
- A prova individual de escolha múltipla servirá para avaliar todos o objetivos de aprendizagem (OA1 a OA8), de uma forma individual;
- O projeto integrador servirá para avaliar todos os objetivos de aprendizagem de uma forma mais prática.

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

- Topics on the syllabus will be exposed during lectures and further discussed using examples.
- During the first part of the semester, students will complete hands-on exercises that will cover the learning objectives from LO1 to LO7. These exercises are presented in the form of tutorials in which students are driven, not only to solve problems, but also to understand why these exercises are solved as suggested.
- In the second part of the semester, students will develop an integration project that will re-exercise the learning objectives from LO1 to LO7, but also LO8.

**Evaluation:**

- Participation assessment in practical classes will help students to engage in solving the proposed exercises so that they fulfill their pedagogical function;
- The individual computer test will assess the learning objectives LO2 and part of LO3;
- The individual multiple choice test will serve to evaluate all learning objectives (LO1 to LO8) individually;
- The integrative project will serve to evaluate all learning objectives in a more practical fashion.

**4.4.9. Bibliografia de consulta/existência obrigatória:**

- Bruce, E. (2010). *Thinking in Java*.
- Miles, R., & Hamilton, K. (2006). *Learning UML 2.0*. O'Reilly Media, Inc..
- Beck, K. (2003). *Test-driven development: by example*. Addison-Wesley Professional.
- Gamma, E. (1995). Helm, R., Johnson, R., Vlissides, J.: *Design patterns: elements of reusable object-oriented software*. Addison Wesley Longman, Inc, January, 1(5), 1.
- Fowler, M. (2018). *Refactoring: improving the design of existing code*. Addison-Wesley Professional.
- Martin, R. C. (2002). *Agile software development: principles, patterns, and practices*. Prentice Hall.
- Visser, J., Rigal, S., Wijnholds, G., van Eck, P., & van der Leek, R. (2016). *Building Maintainable Software, C# Edition: Ten Guidelines for Future-Proof Code*. O'Reilly Media, Inc.

**Mapa IV - Linguagens e Tecnologias Web****4.4.1.1. Designação da unidade curricular:**

*Linguagens e Tecnologias Web***4.4.1.1. Title of curricular unit:***Web Languages and Technologies***4.4.1.2. Sigla da área científica em que se insere:***EINFC***4.4.1.3. Duração:***Semestral***4.4.1.4. Horas de trabalho:***162***4.4.1.5. Horas de contacto:***52 (26T + 26TP)***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):***André Monteiro de Oliveira Restivo (T - 26h - 2 turmas; TP - 52h - 10 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***José Paulo de Vilhena Geraldês Leal (T - 26h - 2 turmas; TP - 26h - 10 turmas)**João António Correia Lopes (TP - 52h - 10 turmas)**Sérgio Sobral Nunes (TP - 52h - 10 turmas)**Docentes a contratar (TP - 78h - 10 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Nesta unidade curricular pretende-se que os estudantes desenvolvam competências nas principais linguagens e tecnologias web. No final desta unidade curricular o estudante deverá ser capaz de:*

- 1. Estruturar documentos usando a linguagem HTML;*
- 2. Usar a linguagem CSS para desenhar e dispor elementos numa página web;*
- 3. Desenvolver páginas web dinâmicas, com ligação a bases de dados relacionais, usando as linguagens PHP e SQL;*
- 4. Usar a linguagem JavaScript de forma a melhorar a interação entre o utilizador e a página web;*
- 5. Perceber o funcionamento do protocolo HTTP e desenvolver sistemas usando uma arquitetura REST;*
- 6. Analisar e resolver os vários tipos de problemas de segurança associados ao desenvolvimento web;*
- 7. Usar expressões regulares para validar e preparar dados em formato de texto;*
- 8. Perceber a relação entre a linguagem XML e o desenvolvimento web, assim como as várias tecnologias associadas (e.g. XPath).*

**4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***The main objective of this course is to allow students to develop skills in the major web languages and technologies.**At the end of this course the student should be able to:*

- 1. Structure documents using the HTML language;*
- 2. Use CSS language to draw and layout elements on a web page;*
- 3. Develop dynamic web pages, that connect to relational databases, using the PHP and SQL languages;*
- 4. Use the JavaScript language to improve web page interaction with the user;*
- 5. Understand the operation of the HTTP protocol and develop systems using a REST architecture;*
- 6. Analyze and resolve the various types of security issues associated with web development;*
- 7. Use regular expressions to validate and prepare data in text format;*
- 8. Understand the relationship between XML language and web development, as well as the various associated technologies (eg. XPath).*

**4.4.5. Conteúdos programáticos:***1. HTML: história, estrutura, elementos e marcas;**2. CSS: história, seletores, propriedades e valores. Posicionamento, especificidade e herança. Design responsivo e adaptável;*

3. *PHP: história, variáveis, estruturas de controle, matrizes e funções. Classes, parâmetros HTTP e acesso a bases de dados usando o PDO;*
4. *java script: variáveis, estruturas de controle, funções, objetos, matrizes e DOM. Conceitos avançados sobre classes e Ajax;*
5. *Segurança web: tipos de ataques e vulnerabilidades. Prevenção de ataques do tipo XSS, CSRF e SQL injection. Armazenamento de senhas de acesso;*
6. *Expressões regulares: noções básicas e uso em PHP, Javascript e HTML;*
7. *HTTP: pedidos, respostas e métodos. Arquiteturas REST;*
8. *Desenvolvimento de progressive web apps (PWAs) usando APIs HTML;*
9. *XML: história, XML válido, namespaces, tecnologias e aplicações;*
10. *XPath: tipos de dados, caminhos de localização, eixos, testes de nós, predicados e abreviações.*

#### 4.4.5. Syllabus:

1. *HTML: history, structure, elements and tags;*
2. *CSS: history, selectors, properties, and values. Positioning, specificity and inheritance. Responsive and adaptable design;*
3. *PHP: history, variables, control structures, matrices and functions. Classes, HTTP parameters and database access using PDO;*
4. *java script: variables, control structures, functions, objects, matrices and DOM. Advanced concepts about classes and Ajax;*
5. *Web security: types of attacks and vulnerabilities. Prevention of XSS, CSRF, and SQL injection attacks. Password storage;*
6. *Regular expressions: basic usage in PHP, Javascript and HTML;*
7. *HTTP: requests, responses, and methods. REST architectures;*
8. *Development of progressive web apps (PWAs) using HTML APIs;*
9. *XML: history, valid XML, namespaces, technologies, and applications;*
10. *XPath: Data types, location paths, axes, node tests, predicates, and abbreviations.*

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

*Existe um mapeamento direto entre os conteúdos programáticos e os objetivos de aprendizagem. A unidade curricular tem como objetivo central fazer uma introdução às principais tecnologias web. Para atingir este objetivo, os conteúdos são organizados no programa de acordo com três tópicos: os conceitos e tecnologias para o desenho e implementação de páginas web (pontos 1 a 4 e 7 a 8 do programa); a segurança web (ponto 5 do programa); e as tecnologias XML para estruturação de dados (pontos 6, 9 e 10 do programa).*

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

*There is a direct mapping between the syllabus and the learning objectives. The main objective of this course is to introduce the main web technologies. To achieve this goal, the contents are organized in the program according to three topics: the concepts and technologies for the design and implementation of web pages (points 1 to 4 and 7 to 8 of the program); web security (point 5 of the program); and XML technologies for data structuring (program points 6, 9 and 10).*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*Nas aulas teóricas são expostos os conceitos com recurso a exemplos práticos sempre que necessário.*

*Nas aulas práticas, na primeira parte do semestre, serão propostos exercícios práticos para serem resolvidos pelos estudantes, em complemento aos conceitos apresentados previamente nas aulas teóricas. Na segunda parte, os estudantes irão desenvolver um projecto onde irão aplicar os conceitos apresentados nas aulas.*

*Avaliação distribuída com exame final.*

*Condições de Frequência: Não exceder o número limite de faltas, realizar um projeto durante o semestre e uma prova final escrita.*

*Fórmula de avaliação:*

*- Nota Final = 0.5 x Projeto + 0.5 x Exame*

*Nota mínima de 8 valores (em 20) no Projeto e Exame*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Lecture classes are used to explain the main concepts present on the syllabus, using practical examples whenever necessary.*

*In the practical classes, exercises about the concepts presented in the lecture classes will be proposed in the first part of the semester. In the second part, students will have to develop a project, in which they are expected to apply the concepts presented in class.*

*Distributed evaluation with a final exam.*

*Terms of frequency: Do not exceed the limit number of absences, do a project during the semester and a final written*

exam.

*Formula of evaluation:*

- Final Grade = 0.5 x Project + 0.5 x Exam

Minimum grade of 8 marks (out of 20) on the Project and the Exam.

**4.4.8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular:**  
*As aulas teóricas são utilizadas para a apresentação e discussão dos tópicos centrais da unidade curricular de acordo com o programa apresentado. Durante estas sessões são usados exemplos práticos para ilustrar problemas e casos reais de aplicação das tecnologias abordadas.*

*Nas aulas teórico-práticas é feita uma revisão inicial dos principais conceitos a aplicar durante a sessão, em seguida o trabalho dos estudantes é orientado por guiões laboratoriais para a aplicação dos conceitos abordados. O recurso a guiões permite a exploração e aplicação dos conceitos e tecnologias de forma individual (no ritmo e na profundidade).*

*Durante as aulas teórico-práticas é feito o acompanhamento do projeto final por parte do docente. Este modelo permite acompanhar o uso das diversas tecnologias através de apresentações individuais, discussões orientadoras, e validação das soluções adotadas.*

**4.4.8. Evidence of the coherence between the teaching methodologies and the intended learning outcomes:**  
*The lectures are used for the presentation and discussion of the core topics of the course according to the syllabus presented. During these sessions practical examples are used to illustrate problems and real cases of application of the technologies covered.*

*In the theoretical-practical classes, an initial review is made of the main concepts to be applied during the session, followed by the students' work guided by laboratory guides for the application of the concepts covered. The use of guides allows the exploration and application of concepts and technologies individually (in rhythm and depth).*

*During the theoretical-practical classes, the final project is monitored by the teacher. This model allows to monitor the use of the various technologies through individual presentations, guiding discussions, and validation of the solutions adopted.*

**4.4.9. Bibliografia de consulta/existência obrigatória:**

*Castro, E., & Hyslop, B. (2011). HTML5 & CSS3: Visual QuickStart Guide (Visual QuickStart Guides). Peachpit Press.*

*Flanagan, D. (2011). java script: The Definitive Guide. O'Reilly.*

*Moller, A., & Schwartzbach, M. I. (2016). An introduction to XML and web technologies. Pearson.*

#### Mapa IV - Matemática Discreta

**4.4.1.1. Designação da unidade curricular:**

*Matemática Discreta*

**4.4.1.1. Title of curricular unit:**

*Discrete Mathematics*

**4.4.1.2. Sigla da área científica em que se insere:**

*M*

**4.4.1.3. Duração:**

*Semestral*

**4.4.1.4. Horas de trabalho:**

*162*

**4.4.1.5. Horas de contacto:**

*26 T 26 TP*

**4.4.1.6. ECTS:**

*6*

**4.4.1.7. Observações:**

*A Lógica constitui a base de qualquer raciocínio científico e essa é a razão primeira da sua inclusão no 1º ano do curso. Para além disso, no caso da Engenharia Informática, a Lógica tem um interesse direto operacional em múltiplas dimensões da profissão.*

*Os objetivos genéricos desta unidade curricular são o desenvolvimento de competências de raciocínio rigoroso e de*

*técnicas de matemática discreta necessárias em várias áreas da informática, como a resolução de problemas, a criação e análise de algoritmos, a teoria da computação, a representação de conhecimento e a segurança.*

#### 4.4.1.7. Observations:

*Logic forms the basis of any scientific reasoning and that is the main reason for its inclusion in the first year of the program. Furthermore, in the case of a Computer Science program, Logic has direct operational relevance in multiple professional aspects.*

*The generic goals of this curricular unit are the development of skills in rigorous reasoning and in the techniques of discrete mathematics required in several areas of computer science like problem solving, algorithm design and analysis, theory of computing, knowledge representation and security.*

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

*António Mário da Silva Marcos Florido (T - 26h - 2 turmas; TP - 78h - 12 turmas)*

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

*Gabriel de Sousa Torcato David (T - 26h - 2 turmas)*

*Gil Coutinho Costa Seixas Lopes (TP - 52h - 12 turmas)*

*José Maria Corte Real da Costa Pereira (TP - 104h - 12 turmas)*

*Docente a contratar (TP - 78h - 12 turmas)*

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

*As competências a adquirir incluem:*

- (1) representar situações utilizando lógica de primeira ordem e analisá-las quer na perspetiva de modelos quer na da prova;*
- (2) dominar os conceitos básicos de conjuntos, relações, ordens parciais e funções;*
- (3) resolver problemas simples de teoria dos números, em particular na sua aplicação à criptografia;*
- (4) resolver equações de aritmética modular;*
- (5) realizar provas indutivas;*
- (6) formular e resolver problemas através de relações de recorrência.*

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

*The skills to be acquired include:*

- (1) representing situations using first order logic and analyzing them both in the models and the proof perspectives;*
- (2) mastering the basic concepts of sets, relations, partial orders, and functions;*
- (3) solving simple problems of number theory, in particular in its application to cryptography;*
- (4) solving modular arithmetic equations;*
- (5) performing inductive proofs;*
- (6) formulating and solving problems through recurrence relations.*

#### 4.4.5. Conteúdos programáticos:

*Lógica proposicional. Métodos de prova em lógica proposicional.*

*Quantificadores e representação do conhecimento. Métodos de prova em lógica de primeira ordem.*

*Introdução à teoria dos números. Congruências e equações de aritmética modular.*

*Indução e recursão. Relações de recorrência.*

*Conjuntos, relações e ordens parciais. Funções.*

#### 4.4.5. Syllabus:

*Propositional logic. Proof methods in propositional logic.*

*Quantifiers and knowledge representation. Proof methods in first order logic.*

*Introduction to number theory. Congruences and modular arithmetic equations.*

*Induction and recursion. Recurrent relations.*

*Sets, relations, and partial orders. Functions.*

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

*Os conteúdos programáticos estão alinhados com os objetivos de aprendizagem, um a um. A apresentação da lógica de primeira ordem nas perspetivas de tradução da língua natural, modelos e prova, para cada elemento da linguagem, permite valorizar o raciocínio rigoroso, acompanhado de uma semântica apelativa. As aplicações dos princípios da teoria dos números a situações do dia-a-dia ajudam a motivar os estudantes para atingirem as competências pretendidas. A inclusão do método de prova por indução matemática imediatamente antes do estudo das relações de recorrência facilita a compreensão deste e permite a consolidação mútua dos tópicos.*

*O estudo integrado de conjuntos, relações, ordens parciais e funções, incluindo uma introdução à cardinalidade de conjuntos, constitui uma base para a utilização desses conceitos na programação e no desenvolvimento de algoritmos.*

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

*The syllabus is aligned with the learning objectives, on a one-to-one basis. Presenting the first order logic under the perspectives of natural language translation, models, and proof, for each language element, enables enforcing rigorous reasoning along with a catchy semantics.*

*Applying the principles of number theory to everyday situations helps motivate students to achieve the desired skills. The inclusion of the proof by mathematical induction method immediately before the study of recurrence relations facilitates the understanding of this and allows the mutual consolidation of the topics. The integrated study of sets, relationships, partial orders and functions, including an introduction to the cardinality of sets, provides a basis for using these concepts in programming and developing algorithms.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*Nas aulas teóricas apresentam-se os assuntos do programa e discutem-se exemplos de aplicação. Nas aulas práticas faz-se a análise e resolução de problemas que visam desenvolver e testar as competências indicadas, recorrendo ao software de apoio na parte da lógica.*

*Tipo de Avaliação: Avaliação distribuída sem exame final*

*Condições de Frequência: Para obter frequência, o estudante deve obter uma avaliação global superior a 7,5 e não exceder o limite legal de faltas. Ter frequência significa que, em caso de reprovação, no ano seguinte pode estar dispensado das aulas.*

*Fórmula de avaliação: Nota =  $[T1+T2+T3+T4-0.8*\min_{i=1,..,4}(Ti)]/3.2$  ,  $T_i$  - nota do teste  $i$*

#### 4.4.7. Teaching methodologies (including students' assessment):

*In theoretical lectures the syllabus topics are presented and application examples are discussed. Practical lectures are devoted to analyzing and solving problems aiming at developing and testing the above mentioned skills, resorting to support software in the logic topics.*

*Type of evaluation: Distributed evaluation without final exam*

*Terms of frequency: To get attendance certificate, the student must obtain a global assessment of 7,5 and attend the legal number of lectures. The attendance certificate may, in case of failure, release the student from attending classes on the next year.*

*Formula Evaluation: Classification =  $[T1+T2+T3+T4-0.8*\min_{i=1,..,4}(Ti)]/3.2$  ,  $T_i$  - test  $i$  classification*

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

*A divisão entre aulas teóricas e práticas, permite concentrar mais estudantes na fase de apresentação de conceitos. Nas aulas teóricas, os vários assuntos vão sendo apresentados com exemplos ilustrativos e mesmo alguns exercícios. No final de cada tópico é feito um questionário rápido on-line para aferir o grau de compreensão na matéria anterior. A existência de quatro testes, um no final de cada capítulo, visa fornecer rapidamente informação aos estudantes sobre a sua progressão.*

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

*The division among the theoretical classes and the practice with smaller groups, allows for the concentration of large numbers of students in the theoretical lectures. In the theoretical classes, the topics are presented with illustrative examples and even some exercises. In the end of each topic, a fast on-line quiz allows to gauge the understanding of the basic concepts just learned. The existence of four tests, one for each chapter, envisages offering fast feedback to the students on their progression.*

#### 4.4.9. Bibliografia de consulta/existência obrigatória:

*Barker-Plummer, D., Barwise, J., Etchemendy, J., Liu, A., Murray, M., & Pease, E. (2011). Language, proof, and logic (Vol. 2). Stanford, CA: CSLI publications.*

*Barwise, J., Etchemendy, J., Allwein, G., Barker-Plummer, D., & Liu, A. (2000). Language, proof and logic. CSLI publications.*

*Goodaire, E. G., & Parmenter, M.M. (1997). Discrete mathematics with graph theory. Prentice Hall PTR.*

*Johnsonbaugh, R. (1997). Discrete Mathematics (Fourth Edition). Upper Saddle River, NJ: Prentice Hall International, 257-280.*

*R. Scheinerman, E.R. (2013). Mathematics: A Discrete Introduction, 3rd ed.. Brooks/Cole Cengage Learning.*

### Mapa IV - Métodos Estatísticos

#### 4.4.1.1. Designação da unidade curricular:

*Métodos Estatísticos*

#### 4.4.1.1. Title of curricular unit:

*Statistical Methods*

#### 4.4.1.2. Sigla da área científica em que se insere:

*M*

**4.4.1.3. Duração:***Semestral***4.4.1.4. Horas de trabalho:***121.5***4.4.1.5. Horas de contacto:***39 (19.5T + 19.5TP)***4.4.1.6. ECTS:***4.5***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):***Jorge Miguel Milhazes de Freitas (T - 19.5h - 2 turmas; TP - 58.5h - 13 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***António Miguel da Fonseca Fernandes Gomes (T - 19.5h - 2 turmas; TP - 39h - 13 turmas)**Armando Luis Ferreira Leitão (TP - 39h - 13 turmas)**Maria Helena Pinto da Rocha Mena de Matos (TP - 117h - 13 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Garantir que os estudantes adquiram uma visão integrada de conceitos e técnicas básicas da Estatística no âmbito do ciclo de estudos de Engenharia Informática.**No final da unidade curricular, os estudantes devem ser capazes de:*

- utilizar métodos para explorar, sumarizar e apresentar dados;
- utilizar métodos de inferência estatística;
- utilizar os métodos e técnicas anteriores de forma crítica e com autonomia na preparação de decisões.

**4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***This course unit aims to provide students with an integrated vision of the basic concepts and techniques of Statistics.**At the end of this course unit, students should be capable of:*

- using methods to explore, summarize and present data;
- using statistical inference methods;
- using adequate methods and techniques of statistical analysis critically and with autonomy in the preparation of decisions.

**4.4.5. Conteúdos programáticos:**

1. **INTRODUÇÃO:** Dados e Observações. Populações e Amostras. Método de Análise Estatística.
2. **ESTATÍSTICA DESCRITIVA:** Caracterização e Representação de Dados Categóricos, Quantitativos e Bivariados.
3. **PROBABILIDADES:** Experiências aleatórias, Espaços Amostrais e Acontecimentos. Probabilidade, Probabilidade Condicional e Acontecimentos Independentes. Teoremas da Probabilidade Total e de Bayes.
4. **VARIÁVEIS ALEATÓRIAS E DISTRIBUIÇÕES DE PROBABILIDADE:** Variáveis Aleatórias Discretas e Contínuas. Função de Probabilidade, de Densidade de Probabilidade e de Distribuição. Parâmetros Populacionais. Independência, Covariância e Correlação.
5. **PRINCIPAIS DISTRIBUIÇÕES DISCRETAS e CONTÍNUAS:** Binomial, Hipergeométrica, Poisson, Uniforme, Exponencial, Normal, Qui-quadrado, t e F.
6. **AMOSTRAGEM E DISTRIBUIÇÕES AMOSTRAIS**
7. **ESTIMAÇÃO, INTERVALOS DE CONFIANÇA E TESTES DE HIPÓTESES**

**4.4.5. Syllabus:**

1. **INTRODUCTION TO STATISTICS:** Data and Observations. Populations and Samples. Statistical Method.
2. **DESCRIPTIVE STATISTICS:** Summarizing Categorical, Quantitative and Bivariate Data.
3. **PROBABILITIES:** Random Experiments. Sampling Spaces and Events. Probability, Conditional Probability and Independence. Total Probability and Bayes Theorem.
4. **RANDOM VARIABLES AND PROBABILITY DISTRIBUTIONS:** Random Variables. Discrete and Continuous Random Variables. Mass, Density and Cumulative Probability Functions. Population Parameters. Covariance and Correlation.
5. **MAIN DISCRETE AND CONTINUOUS DISTRIBUTIONS:** Binomial, Hypergeometric, Poisson, Uniform, Exponential,

*Normal, Chi-square, t and F Distributions.*

**6. SAMPLING AND SAMPLING DISTRIBUTIONS**

**7. ESTIMATION, CONFIDENCE INTERVALS AND HYPOTHESIS TESTING**

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

*Os conteúdos abordados na UC transmitem aos estudantes uma sólida base sobre técnicas básicas de Estatística, como a Estatística Descritiva, a Teoria da Probabilidade, as Distribuições de Probabilidade e Amostrais, a Estimação Pontual, a Estimação por Intervalo e a Inferência Estatística. Como o objetivo da unidade curricular é transmitir aos estudantes um conjunto de ferramentas que lhes permitam tomar decisões de forma crítica e autónoma o programa proposto cumpre e é coerente com os objetivos estabelecidos.*

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

*The topics covered by the course aim to induce in students a solid knowledge of basic statistics, such as Descriptive Statistics, Probability Theory, Probability and Sampling Distributions, Point Estimation, Confidence Intervals and Statistical Inference. Since the main goal of this course is to provide students with a set of statistical tools to help them on the decision-making process therefore the proposed syllabus fulfils, and it is consistent with the established aims.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*Aulas Teóricas: exposição dos conceitos relativos aos capítulos do programa, acompanhada por exemplos de aplicação e resolução de exercícios selecionados. Aulas Teórico-Práticas: resolução de problemas selecionados e esclarecimento de dúvidas.*

*Tipo de Avaliação: Avaliação distribuída com exame final*

*Fórmula de avaliação: A classificação final (CF) será obtida pela seguinte fórmula:*

$$CF = 0.30 AD + 0.70 EF$$

*AD - Avaliação Distribuída:*

$$AD = 2/3 FA + 1/3 TG$$

*FA - Fichas de Avaliação:*

*- 6 fichas de avaliação (aulas práticas);*

*- classificação das fichas de avaliação é obtida pela média das 4 melhores classificações de cada estudante.*

*TG - Trabalho de Grupo:*

*Trabalho de Grupo de pequena dimensão baseado numa competição online.*

*EF - Exame Final*

*- exame realizado em época de exames.*

*Para aprovação à unidade curricular, para além de uma classificação final não inferior a 10 valores, é requerida a classificação mínima de 7 valores no exame final.*

**4.4.7. Teaching methodologies (including students' assessment):**

*Theoretical classes: presentation of the course unit themes followed by examples and problem solving. Theoretical-practical classes: problem solving and clarification of doubts.*

*Type of evaluation: Distributed evaluation with final exam.*

*Formula of evaluation: The final mark (CF) will be obtained by the following formula:*

$$CF = 0.30 AD + 0.70 EF$$

*AD - Quizzes:*

$$AD = 2/3 FA + 1/3 TG$$

*AD - Quizzes:*

*- 6 quizzes (practical classes);*

*- the quizzes mark (AD) is obtained by the average of the best 4 marks achieved by each student.*

*TG - Teamwork assignment:*

*Small teamwork assignment based on an online competition.*

*EF - Final Exam*

*- written exam.*

*To pass this course, apart from a final grade no less than 10, is required a minimum grade of 7 in the final exam.*

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

*As aulas teóricas e teórico-práticas alinham-se com os objetivos de aprendizagem da unidade curricular. A transmissão de conhecimentos e a discussão sobre estes conhecimentos nas aulas teóricas e a resolução de exercícios e problemas nas aulas teórico-práticas permitem estabelecer as bases de aprendizagem dos estudantes. Os tópicos são abordados de uma forma detalhada e deverão ser consolidados pelos estudantes de forma autónoma. Para completar a formação dos estudantes, será realizado um projeto na área do data mining em forma de competição entre estudantes.*

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

*Lectures and tutorial classes are aligned with the learning objectives of the course. The knowledge transmission and discussion in lectures and solving exercises and problems in tutorial classes lay the foundations for student learning. The topics are detailed covered and should be consolidated by students autonomously. To complete the training of students, a project will be conducted in the area of data mining in the form of competition between students.*

**4.4.9. Bibliografia de consulta/existência obrigatória:**

*Gomes, A.M. & Oliveira, J.F. (2018). Estatística - Apontamentos de Apoio às Aulas.*

*Guimarães, R.C. & Cabral, J.A.S. (2011). Estatística, 2ª edição, Verlag Dashofer.*

*Devore, J. L. (2006). Modern mathematical statistics with applications.*

*Tintle, N., Chance, B.L., Cobb, G.W., Rossman, A.J., Roy, S., Swanson, T. & VanderStoep, J. (2015). Introduction to Statistical Investigations. Wiley.*

*Wonnacott, T. H., & Wonnacott, R. J. (1990). Introductory statistics for business and economics (Vol. 4). New York: Wiley.*

**Mapa IV - Programação Funcional e em Lógica****4.4.1.1. Designação da unidade curricular:**

*Programação Funcional e em Lógica*

**4.4.1.1. Title of curricular unit:**

*Functional and Logic Programming*

**4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

**4.4.1.3. Duração:**

*Semestral*

**4.4.1.4. Horas de trabalho:**

*162*

**4.4.1.5. Horas de contacto:**

*52 (26T + 26TP)*

**4.4.1.6. ECTS:**

*6*

**4.4.1.7. Observações:**

*Esta Unidade Curricular (UC) aborda dois paradigmas de programação baseados em fundamentos matemáticos, nomeadamente a programação funcional fortemente tipada e a programação em lógica. Estes paradigmas preconizam a construção de programas com base em formulações matemáticas tornando mais fácil a especificação de restrições de domínio por construção e a verificação de correção de soluções usando métodos formais.*

**4.4.1.7. Observations:**

*This course covers two programming paradigms based on mathematical foundations, namely, strongly typed functional programming and logic programming. These paradigms promote building programs based on mathematical principles, making it easier to specify domain constraints by construction and the verification of correctness using formal methods.*

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

*Pedro Baltazar Vasconcelos (T - 26h - 2 turmas; TP - 52h - 11 turmas)*

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

*Rui Carlos Camacho de Sousa Ferreira da Silva (T - 26h - 2 turmas; TP - 26h - 11 turmas)*

*Daniel Augusto Gama de Castro Silva (TP - 78h - 11 turmas)*

*Henrique Daniel de Avelar Lopes Cardoso (TP - 78h - 11 turmas)*

*Vítor Manuel de Moraes Santos Costa (TP - 52h - 2 turmas)*

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

*No final da UC, os estudantes deverão ser capazes de:*

*1. Usar tipos pré-definidos e algébricos de Haskell para representar valores de um domínio específico*

2. Definir transformações genéricas sobre estruturas de dados indutivas (por exemplo: árvores) como funções polimórficas em Haskell
3. Decompor problemas de programação em funções puras em Haskell operando sobre tipos de dados estruturados
4. Provar equivalência entre funções em Haskell usando as definições com equações e indução
5. Representar factos e relações como programas em Prolog
6. Compreender o modelo de execução de programas em Prolog
7. Modelar problemas de pesquisa como programas lógicos e queries em Prolog

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

At the end of the course, students should be able to:

1. Use pre-defined and algebraic types in Haskell to represent values of a specific domain
2. Define generic transformations on inductive data structures (for example: trees) as polymorphic functions in Haskell
3. Decompose programming problems into pure Haskell functions operating over structured data types
4. Prove equivalence of Haskell functions using definitions by equations and induction
5. Represent facts and relations as Prolog programs
6. Understand the execution model of Prolog programs
7. Model search problems as logic programs and queries in Prolog.

#### 4.4.5. Conteúdos programáticos:

##### I. Programação Funcional

- a) Expressões, avaliação e valores. Tipos pré-definidos. Definições usando equações.
- b) Polimorfismo paramétrico e “bounded”. Classes de tipos fundamentais.
- c) Expressões lambda. Aplicação parcial e “currying”. Funções de ordem superior do prelúdio-padrão.
- d) Definição de tipos algébricos; encaixe de padrões e definições recursivas.
- e) Exemplos de programação: árvore equilibradas; árvores sintáticas; layout de texto; visualização e jogos.
- f) Programação com I/O.
- g) Provas de equivalência usando definições por equações e indução.

##### II. Programação em Lógica

- a) Lógica proposicional e de predicados. Cláusulas de Horn. Factos e regras. Termos de Herbrand. Unificação.
- b) Linguagem Prolog. Modelo de execução. Resolução SLD. Negação por falha.
- c) Bases de factos e relações. Programação com recursão e estruturas.
- d) Aritmética. Predicados extra-lógicos e de controlo.
- e) Exemplos de programação: procura, jogos, manipulação simbólica.

#### 4.4.5. Syllabus:

##### I. Functional Programming module

- a) Expressions, reductions and values. Built-in simple and structured types. Definitions using equations.
- b) Parametric and bounded polymorphism. Fundamental type classes.
- c) Lambda expressions. Currying and partial application. Higher-order functions of the standard prelude.
- d) Defining algebraic data types; pattern matching and recursive definitions
- e) Programming examples: search trees; syntax trees; text layout; visualization and games
- f) Programming with I/O.
- g) Equivalence proofs using equational theory and induction.

##### II. Logic Programming module

- a) Propositional and predicate logic. Horn clauses. Facts and rules. Herbrand terms. Unification.
- b) The Prolog language. Execution model. SLD resolution. Negation by failure.
- c) Databases of facts and relations. Programming with recursive structures.
- d) Arithmetic. Extra-logic and control predicates
- e) Examples of logic programs: search, games, symbolic manipulation.

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

Os conteúdos dos pontos I.a) e I.b) contribuem diretamente para atingir os objetivos de aprendizagem 1 e 2. Os conteúdos I.c), I.d) e I.e) contribuem também para o objetivo 2. Os exemplos abordados em I.f) e I.g) bem como a realização de trabalho prático permite aos estudantes atingir o objetivo 3. O objetivo 4 é consolidado com os tópicos abordados em I.h).

Os conteúdos dos pontos II.a) a II.c) introduzem os conceitos fundamentais da programação em lógica e do seu modelo de execução para atingir os objetivos 5 e 6. O objetivo 7 é atingido pelos pontos II.d) e II.e).

Os dois paradigmas lecionados revestem-se de especial importância no âmbito do conjunto de linguagens e paradigmas de programação lecionadas no CE, dada a sua relevância em áreas como a Inteligência Artificial e Ciência e Análise de Dados. As abordagens lecionadas nesta UC contribuem ainda para um maior desenvolvimento das capacidades de raciocínio abstrato por parte dos estudantes.

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

The topics I.a) and I.b) directly contribute to achieving learning objectives 1 and 2. The contents addressed in I.c), I.d) and I.e) also contribute to objective 2. The examples examined in I.f) and I.g) as well as the programming assignment allows students to reach objective 3. Objective 4 is consolidated through topics covered in I.h).

The topics of points II.a) to II.c) introduce the fundamental concepts of logic programming and their execution model to achieve objectives 5 and 6. Objective 7 is achieved by points II.d) and II.e).

The two paradigms taught in this course are particularly relevant in the set of languages and programming paradigms taught in the study cycle, given their relevance in areas such as Artificial Intelligence and Science and Data Analysis.

*The approach presented also contribute to the development of abstract reasoning by students, also contributing to their formative process.*

#### **4.4.7. Metodologias de ensino (avaliação incluída):**

*As aulas teóricas são usadas para exposição de fundamentos, acompanhada da apresentação e discussão de exemplos ilustrativos. As aulas teórico-práticas permitem a sistematização de conhecimentos por meio da resolução de exercícios de programação e acompanhamento da realização dos trabalhos práticos.*

*Tipo de Avaliação: Distribuída sem exame final.*

*Condições de Frequência: Um estudante obtém frequência se não ultrapassar o número máximo de faltas (25% das aulas TP previstas).*

*A avaliação será feita com base em 2 testes e 2 trabalhos práticos (um teste e um trabalho sobre o primeiro módulo e um segundo teste e trabalho sobre o segundo módulo).*

*P1, P2: Classificação dos trabalhos práticos (0-10)*

*T1, T2: Classificação dos testes intermédios (0-10)*

*Para obtenção de aprovação os estudantes devem ter uma classificação mínima de 40% nos dois testes intermédios.*

*Fórmula da classificação final:  $60\% * (T1+P1) + 40\% * (T2+P2)$*

#### **4.4.7. Teaching methodologies (including students' assessment):**

*Theoretical lectures are used to explain fundamentals, accompanied by the presentation and discussion of illustrative examples. Theoretical-practical (TP) classes allow the systematization of knowledge through the resolution of programming exercises and monitoring of progress in student's assignments.*

*Assessment Type: Distributed without final exam.*

*Frequency Conditions: A student obtains frequency if she/he does not exceed the maximum number of absences (25% of the planned TP classes).*

*The evaluation will be based on 2 tests and 2 practical programming assignments (one test and one assignment for the 1st module and a second test and assignment for the 2nd module).*

*P1, P2: Classifications of programming assignments (0-10)*

*T1, T2: Classifications of intermediate tests (0-10)*

*To obtain approval, students must have a minimum rating of 40% in both two intermediate tests.*

*Formula for the final mark:  $60\% * (T1 + P1) + 40\% * (T2 + P2)$*

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

*Esta UC pretende combinar o ensino de fundamentos científicos e metodológicos com uma componente prática de programação. A organização das aulas teóricas e teórico-práticas permite atingir esse objetivo, sendo as aulas teóricas usadas para transmitir os conhecimentos fundamentais, fazendo igualmente uma ligação dos mesmos à componente mais prática e tecnológica da unidade curricular, a qual é desenvolvida nas aulas teórico-práticas, com a resolução de exercícios e a realização de trabalhos práticos.*

*A realização de dois trabalhos práticos, focados nos dois paradigmas de Programação Funcional e Programação em Lógica, respetivamente, permite aos estudantes colocar em prática os conhecimentos adquiridos. Estando esta UC enquadrada no contexto de um curso da área de engenharia, esta prática reveste-se de especial importância, pretendendo dotar os estudantes da capacidade de implementar programas de alguma complexidade.*

*A realização de duas provas de avaliação escritas é também essencial para garantir que a componente teórica e científica da UC não é descurada, permitindo assim avaliar a assimilação de conhecimentos teóricos e práticos por parte dos estudantes.*

*A divisão de ambas as componentes de avaliação (trabalhos práticos e testes escritos) em dois momentos permite avaliar separadamente os conhecimentos adquiridos numa primeira parte da UC, focada em Programação Funcional, dos adquiridos numa segunda parte da UC, focada em Programação em Lógica. A primeira parte, por envolver conceitos transversais a várias linguagens de programação modernas (tipos, modularidade, correção de software), tem um maior peso letivo, refletido na fórmula de avaliação.*

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

*This course aims to combine the teaching of scientific and methodological foundations with a practical programming component. The organization of theoretical and theoretical-practical classes makes it possible to achieve this objective, with the theoretical classes being used to transmit fundamental knowledge while establishing connections to the more practical and technological component of the course. This is developed in the theoretical-practical classes through the resolution of exercises and practical work.*

*The realization of two programming assignments, focused on the taught paradigms, allows students to put the acquired knowledge into practice. Since this course is part of an engineering degree, this practice is of particularly important, aiming to provide students with the ability to implement programs of some complexity.*

*The realization of two written assessment tests is also essential to ensure that the theoretical and scientific component of the course is not neglected, thus allowing to evaluate the assimilation of theoretical and practical knowledge by the students.*

*The division of both assessment components (practical assignments and written tests) in two moments allows to evaluate separately the knowledge acquired in a first part, focused on Functional Programming, from that acquired in a second part, focused on Logic Programming. The first part, because it involves concepts that are relevant to several modern programming languages (types, modularity, software correction), is given a greater weight, reflected in the assessment formula.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

Hutton, G. (2016). *Programming in Haskell (2nd edition)*. Cambridge University Press, ISBN: 978-1316626221.  
 Bird, R. (1998). *Introduction to Functional Programming using Haskell (2nd edition)*. Prentice-Hall Europe. ISBN: 978-0-13-484346-9.  
 Hudak, P. (2000). *The Haskell school of expression: learning functional programming through multimedia*. Cambridge University Press. ISBN: 978-0-521-64338-2.  
 Sterling, L. and Shapiro, E. (1994). *The Art of Prolog: Advanced Programming Techniques (second edition)*. MIT Press. ISBN: 978-0-262-69163-5.

#### Mapa IV - Programação

##### 4.4.1.1. Designação da unidade curricular:

*Programação*

##### 4.4.1.1. Title of curricular unit:

*Programming*

##### 4.4.1.2. Sigla da área científica em que se insere:

*EINFC*

##### 4.4.1.3. Duração:

*Semestral*

##### 4.4.1.4. Horas de trabalho:

*162*

##### 4.4.1.5. Horas de contacto:

*52 (26T + 26TP)*

##### 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):

*Jorge Alves da Silva (T - 26h - 2 turmas; TP - 52h - 12 turmas)*

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

*Ana Paula Nunes Gomes Tomás (T - 26h - 2 turmas; TP - 52h - 12 turmas)*

*Pedro Gabriel Dias Ferreira (TP - 52h - 12 turmas)*

*Rui Carlos Camacho de Sousa Ferreira da Silva (TP - 52h - 12 turmas)*

*Docente(s) a contratar (TP - 104h - 12 turmas)*

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

*Os objetivos principais desta unidade curricular são dotar os estudantes com:*

- conhecimentos fundamentais sobre a programação procedimental e a programação baseada em objetos;*
- a capacidade de aplicar esses paradigmas de programação para desenvolver programas, usando a linguagem C/C++ como ferramenta.*

*Os estudantes que concluírem com sucesso esta unidade curricular, deverão ser capazes de:*

- Resolver problemas de programação de baixa/média complexidade utilizando a linguagem C/C++ e as abstrações definidas na biblioteca standard (STL) de C++;*
- Desenvolver soluções para problemas de programação que passem pela definição e implementação em C/C++ de abstrações definidas pelo próprio;*
- Escrever programas bem estruturados, legíveis e bem comentados.*
- Compilar programas, executá-los e corrigir os seus erros.*

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

*The main objectives of this curricular unit are to transmit to the students:*

- fundamental knowledge on procedural and object oriented programming techniques;*
- the ability of applying those programming paradigms to develop programs, using the C/C++ language as development*

tool.

*The students who complete successfully this curricular unit must be able:*

- To solve programming problems of low/medium complexity using the C/C++ programming language and the abstractions from the C++ standard template library (STL);
- To develop solutions to programming problems by defining and implementing user-defined abstractions in C/C++;
- To write well structured, legible and well commented programs;
- To compile programs, execute and debug them.

#### 4.4.5. Conteúdos programáticos:

1. **INTRODUÇÃO À LINGUAGEM C/C++.** Tipos de dados simples e E/S básica. Operadores. Expressões. Instruções de controlo de fluxo. Tipos de dados estruturados. Apontadores e alocação dinâmica de memória. Funções: passagem de parâmetros e retorno de valores; funções recursivas; "overloading"; "templates". "Streams"; "stringstreams".
2. **INTRODUÇÃO ÀS CLASSES E ABSTRAÇÃO DE DADOS.** Conceitos fundamentais: encapsulamento; membros-dado e membros-função; visibilidade dos membros; construtores. "Templates". Destrutores. Construtores de cópia. Sobrecarga de operadores. Introdução à conceção de programas OO.
3. **STANDARD TEMPLATE LIBRARY (STL).** Contentores (tipos principais), iteradores e algoritmos.
4. **HERANÇA E POLIMORFISMO.** Conceitos fundamentais. Classes derivadas. Formas de polimorfismo. Classes abstratas.
5. **COMPILAÇÃO SEPARADA. "NAMESPACES". TRATAMENTO DE EXCEÇÕES.**
6. **EXEMPLOS DE APLICAÇÃO.**

#### 4.4.5. Syllabus:

1. **INTRODUCTION TO C/C++.** Simple data types and basic I/O. Operators. Expressions. Flow control: selection and repetition. Structured data types. Pointers and dynamic memory allocation. Functions: parameter-passing mechanisms and return values; recursive functions; overloading; templates. Streams; stringstream.
2. **INTRODUCTION TO CLASSES AND DATA ABSTRACTION.** Fundamental concepts: encapsulation; data-members and function-members; constructors. Destructors. Copy-constructors. Operator overloading. Template classes. Introduction to the design of object-oriented programs.
3. **STANDARD TEMPLATE LIBRARY (STL).** Containers (fundamental types), iterators and algorithms.
4. **INHERITANCE AND POLYMORPHISM.** Fundamental concepts. Derived classes. Forms of polymorphism. Abstract classes.
5. **SEPARATE COMPILATION. NAMESPACES. EXCEPTION HANDLING.**
6. **APPLICATION EXAMPLES.** Program conception using procedural and object oriented design (along the curricular unit).

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

*Os conteúdos programáticos exploram os principais conceitos da programação procedimental e da programação orientada por objetos, com recurso à linguagem C/C++. No ponto 1 são introduzidos os conceitos básicos da programação procedimental, permitindo aos estudantes desenvolver programas estruturados que usem diferentes tipos de dados, simples e estruturados, recorrendo a alocação de memória estática ou dinâmica. Neste ponto são já introduzidos alguns dos contentores da STL que serão analisados com maior detalhe no ponto 3. Nos pontos 2 e 4 são formalmente introduzidos os conceitos fundamentais sobre programação orientada por objetos, nomeadamente, abstração de dados, encapsulamento, herança e polimorfismo, alguns dos quais já foram sendo aflorados nos pontos anteriores. No ponto 5 são abordados com maior profundidade alguns temas que vão sendo introduzidos ao longo do curso, em particular o tratamento de exceções, importante para o desenvolvimento de programas robustos.*

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

*The syllabus explores the main concepts of procedural programming and object oriented programming, using the C / C ++ language. In point 1 the basic concepts of procedural programming are introduced, allowing students to develop structured programs that use different types of data, simple and structured, using static or dynamically allocated memory. In this point, some STL containers are already introduced, which will be analyzed in more detail in point 3. In points 2 and 4, the fundamental concepts of object-oriented programming, namely data abstraction, encapsulation, inheritance and polymorphism, some of which have already been outlined in the previous points, are formally introduced. In point 5, some topics that have been introduced throughout the course are discussed in more depth, particularly exception handling, important for the development of robust programs.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*Aulas Teóricas: exposição dos temas programáticos acompanhada por exemplos de aplicação; discussão e desenvolvimento de código.*

*Aulas Teórico-Práticas: resolução de exercícios de programação focados nos temas tratados nas aulas teóricas; esclarecimento de dúvidas sobre os trabalhos práticos, a realizar por grupos de 2 estudantes, fundamentalmente em períodos extra aula.*

*Tipo de Avaliação: distribuída com exame final.*

*Condições de Frequência: Condições para obtenção de frequência: não exceder o limite de faltas e obter uma classificação mínima de 40% em AvD de 30% em MT (ver abaixo).*

*Fórmula de avaliação: Avaliação distribuída (AvD) com exame final (AvE).*

- AvD - a classificação é obtida com base num "mini-teste" (MT), a realizar em meados do semestre, e do segundo dos

*2 trabalhos práticos propostos (TP2).*

- $AvD = MT * 40\% + TP2 * 60\%$
- *Classificação final =  $AvD * 50\% + AvE * 50\%$*
- *É condição de aprovação a obtenção de uma classificação mínima de 40% na componente AvE.*

#### **4.4.7. Teaching methodologies (including students' assessment):**

*Lectures: presentation of programmatic themes accompanied by application examples; discussion and development of code.*

*Practical lessons: resolution of programming exercises focused on the themes exposed in the lectures; tutoring of the projects, to be developed by groups of 2 students, mainly out of classes.*

*Type of evaluation: distributed with final exam.*

*Terms of frequency: Do not exceed the absence limit and obtain a minimum of 40% in AvD and a minimum of 30% in MT (see below).*

*Formula of evaluation: Distributed evaluation (AvD) with final exam (AvE).*

- *AvD - the grading is based on the result obtained in a short exam (MT), that will take place by the middle of the semester, and of the second of the 2nd programming project (TP2).  $AvD = MT * 40\% + TP2 * 60\%$*
- *Final grading =  $AvD * 50\% + AvE * 50\%$*
- *To be approved in this course, a minimum of 40% for the AvE evaluation component is required.*

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

*As aulas teóricas são dedicadas essencialmente à apresentação dos conceitos fundamentais da programação procedimental e da programação baseada em objetos, a qual é feita recorrendo a programas completos e não apenas a extratos de código. Os programas são analisados e são discutidas soluções alternativas que, frequentemente, são testadas em computador, durante as aulas. Estes programas são disponibilizados na página web da UC, sendo os estudantes incentivados a testarem soluções diferentes das apresentadas ou a alterarem os programas no sentido de resolverem problemas semelhantes ou complementares. Para facilitar a apreensão dos conceitos, estes são apresentados de forma gradual, recorrendo frequentemente a um programa que vai evoluindo à medida que um problema a resolver se vai complicando.*

*As aulas teórico-práticas são essencialmente destinadas à resolução de um conjunto de exercícios simples de programação, sendo cada exercício focado num dos temas tratados nas aulas teóricas. O objetivo principal destes exercícios é a aprendizagem da sintaxe da linguagem e dos conceitos básicos da programação, à medida que vão sendo apresentados nas aulas teóricas; é fomentada a discussão entre estudantes sobre os problemas que vão surgindo e as propostas de solução que vão sendo por eles apresentadas.*

*Ao longo do semestre são propostos dois projetos de programação cujo desenvolvimento será feito essencialmente fora das aulas, por grupos de dois estudantes, fomentando assim a capacidade de trabalho em equipa. A evolução dos projetos é avaliada periodicamente, sendo discutidas as soluções propostas pelos estudantes e sugeridas pelo docente as correções que se revelarem necessárias. No final de cada projeto, cada grupo faz uma demonstração do funcionamento do programa, respondendo a questões sobre o código desenvolvido.*

*Ao longo do semestre, os estudantes são incentivados a usar boas práticas de programação, nomeadamente, a escreverem programas bem estruturados, legíveis e bem comentados, e a testarem adequadamente os programas que desenvolvem. Todos estes aspetos são tidos em conta na avaliação dos projetos.*

*A forma como as aulas são conduzidas, os exercícios e os projetos realizados e a forma como estes são avaliados, contribuem para que todos os objetivos de aprendizagem enunciados sejam cumpridos. Através deste conjunto de práticas procura-se estimular o estilo de aprendizagem "aprender fazendo".*

*O regime de avaliação foi concebido para aferir os conhecimentos e competências adquiridas individualmente por cada estudante. Apesar de haver uma componente de avaliação de trabalho de grupo (30% da nota final), a componente principal (70%) é relativa ao desempenho individual (mini-teste=20% e exame=50%). A realização de um miniteste, aproximadamente a meio do semestre, tem como objetivo fomentar o estudo individual.*

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

*Theoretical classes are devoted essentially to the presentation of the fundamental concepts of procedural programming and object-based programming, which is done using complete programs and not just code extracts. The programs are analyzed and alternative solutions, that are often tested on the computer during classes, are discussed. These programs are available on the course website, and students are encouraged to try different solutions than the ones presented or to change the programs to solve similar or complementary problems. To make the concepts easier to grasp, they are presented gradually, often using a program that evolves as a problem to be solved gets complicated. Theoretical-practical classes are essentially aimed at solving a set of simple programming exercises, each exercise focused on one of the topics addressed in the theoretical classes. The main purpose of these exercises is to learn the language syntax and the basic concepts of programming as they are presented in lectures; students are encouraged to discuss the problems that arise and the solutions proposed by them.*

*During the semester, two programming projects are proposed, which will be developed essentially outside the classroom, by groups of two students, thus fostering the ability to work in team. The evolution of the projects is evaluated periodically, discussing the solutions proposed by the students and any necessary corrections are suggested by the teacher. At the end of each project, each group demonstrates how the program works and must answer questions about the developed code.*

*Throughout the semester, students are encouraged to use good programming practices, in particular to write well-structured, readable and well-commented programs, and to adequately test the programs they develop. All these aspects are taken into account in project evaluation.*

*The way the classes are conducted, the exercises and the projects carried out and the way they are evaluated, contribute to the fulfillment of all stated learning objectives. Through this set of practices, a "hands-on" learning style is encouraged.*

*The assessment scheme is designed to measure the knowledge and skills acquired individually by each student. Although there is a group work evaluation component (30% of the final grade), the main component (70%) is related to individual performance (mini-test = 20% and exam = 50%). The mini-test, done approximately at mid-semester, aims to foster individual study.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Savitch, W. & Mock K. (2017). Problem Solving with C++. Pearson.  
Lippman, S. & Lajoie, J. & Moo, B. (2012). C++ Primer. Addison-Wesley Professional.  
Stroustrup, B. (2014). Programming: Principles and Practice Using C++. Addison-Wesley Professional.  
Horstmann, C. (2017). Big C++: Late Objects. John Wiley & Sons Inc.*

### **Mapa IV - Projeto Integrador**

#### **4.4.1.1. Designação da unidade curricular:**

*Projeto Integrador*

#### **4.4.1.1. Title of curricular unit:**

*Capstone Project*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

#### **4.4.1.5. Horas de contacto:**

*OT-13h*

#### **4.4.1.6. ECTS:**

*6*

#### **4.4.1.7. Observações:**

*<sem resposta>*

#### **4.4.1.7. Observations:**

*<no answer>*

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

*Nuno Honório Rodrigues Flores (OT - 32.5h)*

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

*Rita Ribeiro (OT - 19.5h)*

*Para os docentes abaixo a carga letiva na unidade curricular é a seguinte: OT\_6,5h*

*Ademar Aguiar*

*Alexandre Carvalho*

*Alípio Jorge*

*Álvaro Figueira*

*Ana Paiva*

*Ana Rocha*

*Ana Tomás*

*André Restivo*

*António Sousa*

*António Coelho*

*António Soares*

*António Florido*

*António Monteiro*

*Bruno Barreto*

*Carla Lopes*

Carlos Soares  
Daniel Silva  
Eduardo Marques  
Fernando Silva  
Filipe Correia  
Gabriel David  
Gil Gonçalves  
Gilberto Almeida  
Henrique Cardoso  
Hugo Ferreira  
João Lopes  
João Faria  
João Vilela  
João Moreira  
João Santos  
Jorge Silva  
Jorge Barbosa  
José Cruz  
José Leal  
Luís Teixeira  
Luís Reis  
Manuel Correia  
Maria Ribeiro  
Pedro Souto  
Pedro Ferreira  
Pedro Ribeiro  
Pedro Brandão  
Ricardo Rocha  
Rolando Martins  
Rosaldo Rossetti  
Rui Silva  
Rui Rodrigues  
Sabine Broda  
Sandra Alves  
Sérgio Crisóstomo  
Sérgio Nunes  
Verónica Orvalho  
Vítor Costa

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

- (A) *Aplicar um processo de Engenharia de Software ao desenvolvimento completo de um sistema de software real, ao longo do semestre, abrangendo a especificação de requisitos, arquitetura, implementação, integração, teste, documentação e demonstração.*
- (B) *Consolidação de conhecimentos fundamentais e experiência sobre a aplicação de algoritmos, técnicas e práticas de desenvolvimento de software, em contextos específicos de domínio de aplicação.*
- (C) *Utilizar e adequar ferramentas de desenvolvimento de software que permitam o acompanhamento do desenvolvimento do produto durante todo o seu ciclo de vida.*
- (D) *Desenvolver trabalho colaborativo, integrando outros intervenientes de projeto em decisões de desenho e planeamento e delegar, negociar e rever estas decisões em grupo.*
- (E) *Adquirir um elevado grau de autonomia e autodisciplina, assim como competências adequadas de comunicação.*

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

- (A) *To apply a Software Engineering process to the complete development of a real software system, along the curricular unit, covering the specification of requirements, software architecture and design, coding, integration, test, documentation and demonstration.*
- (B) *To consolidate fundamental knowledge and experience about applying algorithms, practices and techniques within software development, at specific application domain contexts.*
- (C) *To use and tailor software development tools that enable the continuous monitoring and tracking of the project or product along its lifecycle.*
- (D) *To develop collaborative work, integrating other project participants in design decisions, planning, delegation, negotiation, and group review.*
- (E) *To acquire a high level of autonomy and self-discipline, as well as adequate communication skills.*

#### **4.4.5. Conteúdos programáticos:**

*Nesta u.c, em paralelo com as restantes UCs do semestre, os estudantes, organizados em equipas de média dimensão (4 a 8 estudantes), devem realizar um projeto de engenharia informática com impacto social, para resolução de um problema real, com base em propostas oriundas preferencialmente de organizações externas à FEUP (empresas, instituições, organismos de solidariedade social, etc.), envolvendo deslocação ao exterior para contato com proponentes e utilizadores finais. No final do semestre, é organizado um evento (feira) de exposição pública dos resultados alcançados. Durante todo o decorrer do projeto, os estudantes serão acompanhados por um docente que orientará a equipa na persecução dos objetivos de aprendizagem.*

**4.4.5. Syllabus:**

*In this course, students will tackle a social-impact, real-world project, developing a software solution grouped in medium-size teams (4 to 8 elements). These projects will be, preferably, proposed by external (outside of FEUP) organizations (companies, social institutions, etc.), having the students to visit those organizations for direct contact with the stakeholders and end-users. At the end of the semester, a final public event ("fair") will take place for demonstration the achieved results. Throughout the entire project lifespan, each team will be monitoring and advised by a member of the faculty staff, ascertaining the persecution of the learning outcomes. As an alternative, and authorized by the programme director, students may propose to undertake an internship in an industrial environment or at a R&D unit (tutored by a faculty staff member) or to participate in a multidisciplinary project, supervised by a faculty staff member.*

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

*A realização de um projeto de desenvolvimento de uma solução de software em equipa, agrega, no seu processo (A), todos os meios para que os objetivos de aprendizagem sejam alcançados. O domínio de aplicação específico do projeto obriga à adequação e aplicação de competências adquiridas ao longo do curso (B). O trabalho em equipa fomenta a colaboração, comunicação, convergência de decisões sobre a solução, e o planeamento e articulação de tarefas (D). O acompanhamento do projeto por um docente, não só garante o sucesso da aprendizagem, mas obriga a equipa a reportar o seu progresso, adequando ferramentas que o permitam (C). Um projeto proposto por um cliente real, coloca a necessidade de, não só comunicar convenientemente, mas articular o planeamento com a disponibilidade e envolvimento do cliente (E). No caso do estudante optar por um estágio ou projeto nos contextos anteriormente referidos, a envolvente do trabalho terá de ser em todo similar ao até agora reportado.*

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

*The context of developing a software project within a team that addresses a real customer's specific needs encompasses all the means to achieve the learning objectives. The specific domain of application enforces the proper identification, adequacy and application of skills, acquired throughout the programme (B). Working in teams fosters collaboration, communication, consensus in decision-making, planning and articulating tasks. (D). Having a faculty staff member monitoring the project, not only enforces progress reporting by the team (through adequate monitoring tools (C)), but also ensures that the learning objectives are being pursued. Dealing with a real customer allows for the team to tackle communication issues and to articulate the project planning and scheduling according to the customer availability and involvement (E). If the student opts for an internship or project (as previously referred), the working environment will have to be similar to the afore-mentioned.*

**4.4.7. Metodologias de ensino (avaliação incluída):**

*As aulas laboratoriais servirão para o acompanhamento e desenvolvimento do projeto, através do uso dos sistemas, linguagens e ferramentas escolhidos para a implementação da solução. O projeto organiza-se em 5 iterações (It), com entregas de artefactos (At), e um peso (P) respetivo na avaliação final.*

*-It: 0 (4 semanas), P: 20%, At: visão do produto, user stories + interfaces utilizador, arquitetura, protótipo vertical funcional.*

*-It: 1-4 (8 semanas), P: 30%, At: plano de desenvolvimento, user stories finais, estimativas, testes unitários e de aceitação.*

*-Entrega final, P: 30%, At: produto, código-fonte documentado, pitch+video+slides demo, documentação técnica final.*

*-Plataforma de projeto montada e o desempenho individual dos estudantes, P:20%.*

*No caso de um estágio em ambiente industrial, ou de um projeto I&D multidisciplinar, será da responsabilidade do tutor/orientador, definir as componentes avaliativas, adequando-as ao contexto, dimensão e complexidade do trabalho.*

**4.4.7. Teaching methodologies (including students' assessment):**

*Laboratory classes are used for the implementation of the application. The development of the project will be divided into five iterations (It), with expected deliverables (De), contributing a percentage (Pr) of the final grade:*

*-It 0 – P: 20%, 4 weeks, De: vision, user stories + user interfaces, architecture, vertical functional prototype.*

*-It 1-4 – P: 30%, 8 weeks, De: development process plan, user stories ready, estimates, acceptance and unit tests.*

*-Finale – P: 30%, De: product release (signup, download, etc.), documented source code repository, pitch + video + slides demo, updated technical reports.*

*The remaining 20% of the final grade evaluate the project management platform and individual performance.*

*If the student opts for an internship or multi-disciplinary R&D project, the tutor/supervisor will be responsible for defining, articulating and evaluating the student's outcomes, which will have to adequately be adjusted to the context, size and complexity of the project.*

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

*A metodologia de ensino com base num projeto permite que os estudantes, não só adquiram e pratiquem os conceitos abordados num ambiente real, mas igualmente desenvolvam o espírito de colaboração, cooperação e trabalho em equipa. O acompanhamento semanal da evolução dos projetos, quer presencial, quer através das ferramentas utilizadas para o efeito, permite uma aferição efetiva da evolução dos conhecimentos e competências adquiridas. Através da metodologia de ensino baseado em projetos, é possível não só aplicar os conceitos já adquiridos, como também aprofundá-los através da prática e integração dos mesmos, com um acompanhamento e monitorização regulares por parte dos docentes. Desta forma, adota-se uma filosofia pedagógica de "aprender, fazendo" que estimula a interiorização dos conhecimentos e expõe os estudantes aos problemas reais que surgem durante um projeto de engenharia informática. Através da demonstração pública dos resultados dos projetos, os estudantes são igualmente estimulados ao conceito de "produto" de software, apelando a uma componente inovadora e viabilizadora da possível*

*atração de investidores interessados num produto real.*

*No caso de um estágio em ambiente industrial ou de um projeto I&D multidisciplinar, será garantido pelo corpo docente que o estudante será integrado num ambiente de trabalho que permita a persecução dos objetivos de aprendizagem.*

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

*Team project-based learning is the adopted methodology in this curricular unit. The laboratory classes will enable the tutoring, supervision and monitoring of the project. Through a real-life software project scenario, the students will be able to apply and consolidate all the intended practices and techniques that a software project encompasses. From its inception to its final product delivery, the students will go through all the phases of development, working in teams and collaborating to achieve a common goal. This "learn by doing" pedagogical philosophy allows for a deeper understanding of the practices and exposes the students to the hurdles and difficulties a real software project might present. This methodology focuses on "product delivery", forcing the students to define a viable process to enable a fully functional, ready to use, final product release. On the way, quality standards need to be met, therefore best practices, methods and tools need to be used so that development enables the achievement of those standards. These quality metrics are set, monitored and enforced by the teachers. Innovation is endorsed through a product "pitch", evaluating if the final product will be prone to investment by potential stakeholders. Considering an internship or a multi-disciplinary R&D project, the faculty staff will guarantee that the student will be integrated in a working environment that will allow for the pursuit of the learning objectives.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Schwaber, K., & Beedle, M. (2002). Agile Software Development with Scrum. Upper Saddle River, NJ, United States: Prentice Hall.*

*Beck, K., & Andres, C. (2004). Extreme Programming Explained. Boston, MA: Pearson Education.*

### **Mapa IV - Projeto UP**

#### **4.4.1.1. Designação da unidade curricular:**

*Projeto UP*

#### **4.4.1.1. Title of curricular unit:**

*Project UP*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*DP*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*40,5*

#### **4.4.1.5. Horas de contacto:**

*T-3h; TP-10h*

#### **4.4.1.6. ECTS:**

*1,5*

#### **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):**

*Manuel Firmino da Silva Torres (T - 3h - 1 turma)*

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

*Ademar Manuel Teixeira de Aguiar (TP - 20h - 9 turmas)*

*João António Correia Lopes (TP - 20h - 9 turmas)*

*João Pedro Pedroso Ramos Santos (TP - 20h - 9 turmas)*

*Jorge Manuel Gomes Barbosa (TP - 30h - 9 turmas)*

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

- Conhecer o campus académico e serviços disponíveis;
- Integrar-se na comunidade académica da UP;
- Identificar as competências transversais e a sua importância na carreira da engenharia;
- Aplicar as competências transversais (soft skills tais como trabalho em equipa, comunicação, ética, etc.);
- Analisar cientificamente um tema / resolver um projeto de dificuldade limitada na área de engenharia informática e computação;
- Elaborar um poster, realizar uma apresentação e produzir um relatório científico, de acordo com as normas da comunidade científica.

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

- To know the academic campus and available services;
- To integrate into the academic community of UP;
- To identify transversal competences and their importance in the engineering career;
- To apply transversal skills (soft skills such as teamwork, communication, ethics, etc.);
- To scientifically analyze a theme / solve a project of limited difficulty in the informatics and computing engineering area;
- To make a poster, a presentation, and a scientific report according to the norms of the scientific community.

**4.4.5. Conteúdos programáticos:**

- Técnicas de apresentação;
  - Comunicação visual (poster);
  - Relatório de engenharia (incluindo pesquisa e referências de bibliografia);
- Será ainda ministrada formação nas áreas de:
- Serviços académicos, recursos informáticos e ferramentas colaborativas (ex: Google Apps e Moodle);
  - Aprendizagem eficaz (incluindo gestão de tempo, métodos de estudo, ética, plágio, etc.)
- A UC inclui ainda um trabalho técnico sobre um tema específico do ciclo de estudo do estudante e formação associada a esse tópico que servirá para o trabalho no tema / projeto em equipa.

**4.4.5. Syllabus:**

- Presentation techniques;
  - Visual communication (poster);
  - Engineering report (including research and bibliographic references);
- Training will also be provided in the areas of:
- Academic services, IT resources and collaborative tools (e.g. Google Apps and Moodle);
  - Effective learning (including time management, study methods, ethics, plagiarism, etc.)
- The course also includes technical work on a specific subject of the student course and training associated with that topic that will serve for work on the theme/team project.

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

Na UC de Projeto UP, os conteúdos programáticos foram definidos no sentido de permitirem a realização dos objetivos. A partir da revisão da literatura e de uma experiência de décadas de trabalho nestas áreas, identificaram-se diversos conteúdos para o programa como essenciais para a integração dos estudantes e o desenvolvimento de competências transversais. Assim, os conteúdos foram organizados em diversas áreas conforme as respetivas competências: 1) Integração: conhecer os serviços e recursos académicos, realizar dinâmicas de grupo e atividades de associação à comunidade académica da UP; 2) Competências transversais - trabalho em equipa; comunicação, ética, etc.; análise científica de um tema e/ou resolução de um projeto na área de engenharia informática e computação, com produção de três elementos de comunicação (apresentações, pôsteres, relatórios) para a realização de um congresso final.

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

In the Project UP curricular unit, the syllabus was defined to allow the achievement of the objectives. Based on the literature review and the experience of decades of work in these areas, several contents for the program were identified as essential for the integration of students and the development of transversal competences. Thus, the contents were organized in several areas according to their respective competences: 1) Integration: to know the academic services and resources, perform group dynamics and UP community association activities; 2) transversal skills - teamwork; communication, ethics, etc.; scientific analysis of a project theme and / or resolution in the informatic and computing engineering area, with the production of three communication elements (presentations, posters, reports) for a final congress.

**4.4.7. Metodologias de ensino (avaliação incluída):**

- 1) Semana inicial de formação intensiva dedicada à receção dos estudantes e formação presencial, complementada com e-learning, sobre os vários conteúdos programáticos. Esta semana inclui diversas palestras (métodos expositivo e interrogativo) e módulos de formação prática (métodos demonstrativo e ativo), sendo as respetivas atividades avaliadas num miniteste.
- 2) Durante as restantes semanas realiza-se um trabalho de projeto em equipa. Este trabalho técnico elaborado com rigor científico sobre conteúdos adotados de outras UCs do ciclo de estudos, inclui a realização de um relatório e de um póster, assim como uma apresentação num Congresso Final com jurados do corpo docente. É alvo de avaliação a forma e o conteúdo do relatório realizado, do poster produzido e ainda da respetiva apresentação oral e defesa do

*projeto (questões do júri e/ou da audiência). Este trabalho é apoiado sistematicamente por um monitor e acompanhado por um supervisor que também avalia os trabalhos realizados.*

#### **4.4.7. Teaching methodologies (including students' assessment):**

*1) The initial week of intensive training dedicated to student reception and face-to-face training, complemented with e-learning, on the various syllabus. This week includes several lectures (expository and interrogative methods) and training modules (demonstrative and active methods), being these activities evaluated in a mini-test.  
2) During the next weeks, the students develop one team project work. This technical work, developed with scientific rigor, on content adopted from other UCs of the course includes a report, a poster as well as a presentation at the Final Congress with professors as jurors. The form and content of the report, the poster and the oral presentation and discussion of the project (questions of the jury and/or audience) are evaluated. This work is systematically supported by a monitor and a supervisor who also evaluate the performed work.*

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

*A metodologia está relacionada com o desenho dos objetivos de aprendizagem desta UC definidos em três níveis: saber, saber-fazer e saber-estar. Assim, os métodos expositivo e interrogativo são essencialmente usados para transmitir conhecimentos e incentivar nos estudantes uma análise e reflexão crítica sobre esses conhecimentos, adotando modos de trabalho pedagógico de tipo transmissivo e incitativo, promovendo estratégias de aprendizagem baseadas nos modelos comportamentalista e cognitivista. Os métodos demonstrativo e ativo são sobretudo usados para ensinar técnicas práticas, desenvolver competências e estimular a apropriação do conhecimento, utilizando um modo de trabalho pedagógico de tipo apropriativo e fomentando estratégias de aprendizagem baseadas nos modelos construtivista e sócio-construtivista. Com estas metodologias de ensino pretende-se que os estudantes possam atingir os resultados de aprendizagem desejados e que estejam capazes de utilizar as suas competências pessoais e interpessoais, em situações informais e em contextos académicos e profissionais onde ocorrem atividades inerentes à engenharia.*

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

*The methodology is related to the design of the learning objectives of this course, defined on three levels: knowledge, know-how, and know-being. Thus, the lecture and the interrogative methods are essentially used to transmit knowledge and encourage students in critical analysis and reflection on such knowledge, adopting pedagogical work modes of transmissive and incentive types, promoting learning strategies based on behavioral and cognitive models. Demonstrative and active methods are particularly used to teach practical techniques, develop skills and stimulate knowledge appropriation, using a pedagogical work mode of appropriative type and fostering learning strategies based on constructivist and socio-constructivist models.  
With the use of these teaching methodologies, we intend to allow students to achieve the desired learning outcomes and to be able to use their personal and interpersonal skills, in informal situations and in academic and professional contexts where there are activities related to engineering.*

#### **4.4.9. Bibliografia de consulta/existência obrigatória:**

*Van Emden, J., & Becker, L. (2016). Presentation skills for students. Macmillan International Higher Education.  
Tufte, E. R. (2001). The visual display of quantitative information (Vol. 2). Cheshire, CT: Graphics press.  
Weyers, J. (2007). How to write dissertations and project reports. Pearson Education UK.  
Cornelsen, J. M. (2012). Escrever... com normas: guia prático para elaboração de trabalho. Imprensa da Universidade de Coimbra/Coimbra University Press.  
Hofmann, A. H. (2014). Scientific writing and communication: papers, proposals, and presentations. Oxford Univ. Press.  
Neves, J., Garrido, M., & Simões, E. (2006). Manual de competências pessoais, interpessoais e instrumentais—teoria e prática. Lisboa: Edições Sílabo.  
Caldeira, B. R., & Ferreira, N. A. (2012). Faz o Curso na Maior. Lua de Papel.  
Webster, G. (2017). The Chicago Manual of Style.*

### **Mapa IV - Redes de Computadores**

#### **4.4.1.1. Designação da unidade curricular:**

*Redes de Computadores*

#### **4.4.1.1. Title of curricular unit:**

*Computer Networks*

#### **4.4.1.2. Sigla da área científica em que se insere:**

*EINFC*

#### **4.4.1.3. Duração:**

*Semestral*

#### **4.4.1.4. Horas de trabalho:**

*162*

**4.4.1.5. Horas de contacto:**

52 (26T + 26TP)

**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):**

Manuel Alberto Pereira Ricardo (T - 26h - 2 turmas; TP - 26h - 10 turmas)

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

Rui Pedro de Magalhães Claro Prior (T - 26h - 2 turmas; TP - 78h - 10 turmas)

Ana Cristina Costa Aguiar (TP - 52h - 10 turmas)

Maria Teresa Magalhães da Silva Pinto de Andrade (TP - 104h - 10 turmas)

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

*Redes de Computadores (RCOM) introduz os estudantes no domínio das redes de comunicações. Os conceitos fundamentais endereçados incluem: canais de comunicação e controlo da ligação de dados, modelos de erro e atraso, comunicações multi-acesso, encaminhamento, controlo de fluxo, controlo de congestionamento e segurança em redes. As principais tecnologias discutidas são a Ethernet comutada, a Wireless LAN 802.11, a Internet e a pilha de comunicações TCP/IP. Os projetos a desenvolver em RCOM incluem um protocolo de ligação de dados com controlo de fluxo, uma aplicação a funcionar em modo cliente, a configuração e teste de redes Ethernet e IP, e análise de tráfego.*

*Depois de ter frequentado RCOM o estudante deverá ser capaz de: a) perceber e explicar os conceitos fundamentais das redes de comunicações; b) escolher, comparar ou empregar as tecnologias de redes de comunicações atuais; c) implementar, configurar e testar mecanismos e componentes essenciais de uma rede de comunicações.*

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

*Computer Networks (RCOM) introduces students to the knowledge domain of communication networks. The fundamental concepts addressed include the following: communication channels and data link control, error and delay models, multi-access communications, routing, flow control, congestion control, and network security. The key technologies discussed in RCOM are switched Ethernet, 802.11 Wireless LAN, the Internet, and the TCP / IP communications stack. The projects developed by students include a flow-control protocol, a client application, Ethernet and IP network configuration and testing, and traffic analysis.*

*After attending RCOM the students should be able to: a) understand and explain the fundamental concepts of communication networks; b) choose, compare or employ current communications network technologies; c) implement, configure and test mechanisms and essential components of a communications network.*

**4.4.5. Conteúdos programáticos:**

*Introdução às redes de computadores. Técnicas de transmissão de dados. Controlo da ligação de dados. Modelos para análise de desempenho. Controlo do acesso ao meio. LANs. A Internet. Encaminhamento. Controlo de congestionamento. Aplicações. Segurança em redes.*

**4.4.5. Syllabus:**

*Introduction to computer networks. Transmission and data communication techniques. Data link control. Models for performance evaluation. Medium access control. LANs. Internet. Routing. Congestion control. Applications. Network Security.*

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

*Os conceitos fundamentais das redes de comunicações são endereçados na generalidade dos conteúdos programáticos indicados. As tecnologias de redes de comunicações são endereçadas em 2 conteúdos programáticos (LANs e Internet) e exercitadas em laboratório. As componentes de redes de comunicações implementadas ou configuradas pelos estudantes incluem ligações de dados, configuração de redes IP, aplicações e análise de tráfego.*

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

*The fundamental concepts of communication networks are addressed in most of the indicated syllabus topics. Communication network technologies are addressed in 2 syllabus topics (LANs and Internet) and exercised in the*

*laboratory. Communications network components implemented or configured by students include data link connections, IP network configuration, applications, and traffic analysis.*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*As aulas teóricas serão usadas para apresentação e discussão dos conceitos fundamentais, discussão de casos e resolução de problemas. Nas aulas teórico-práticas serão realizados dois trabalhos laboratoriais: (1) desenvolvimento de uma pilha de comunicações simples; (2) implementação de rede de comunicações com VLANs e routers, desenvolvimento de aplicação simples e análise de tráfego.*

*Tipo de Avaliação: Avaliação distribuída com exame final*

*Condições de Frequência: Notas atribuídas de 0 a 20 valores.*

*L1 - nota do 1º trabalho laboratorial*

*L2 - nota do 2º trabalho laboratorial*

*H - nota dos trabalhos de casa*

*FQ - CLASSIFICAÇÃO DE FREQUÊNCIA*

*FQ = 0,4\*L1 + 0,4\*L2 + 0,2\*H*

*se ( FQ < 8,0 ) FQ = "Reprovado por Falta Frequência"*

*Fórmula de avaliação: Notas atribuídas de 0 a 20 valores.*

*E - nota do exame final*

*FQ - NOTA DE FREQUÊNCIA*

*CF - CLASSIFICAÇÃO FINAL*

*CF = 0,4\*FQ + 0,6\*E*

*se ( E < 8,0 ) CF = E*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Theoretical classes are dedicated to the formal exposition of contents, including problems' solving to illustrate and discuss cases. In laboratory classes the students will carry out two assignments: (1) development of a simple communications stack; (2) implementation of a communications network with VLANs and routers, development of a simple application and traffic analysis.*

*Type of evaluation: Distributed evaluation with final exam*

*Terms of frequency: Grades given in a scale from 0 to 20 values.*

*L1 - grade of 1st lab*

*L2 - grade of 2nd lab*

*H - grade of homework*

*FQ - GRADE OF FREQUÊNCIA*

*FQ = 0,4\*L1 + 0,4\*L2 + 0,2\*H*

*if ( FQ < 8,0 ) FQ = "No Admission to Exams"*

*Formula of evaluation: Grades given in a scale from 0 to 20 values.*

*E - grade of final exam*

*FQ - GRADE OF FREQUÊNCIA*

*CF - FINAL GRADE*

*CF = 0,4\*FQ + 0,6\*E*

*if ( E < 8,0 ) CF = E*

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

*As aulas teóricas servem para introduzir os tópicos do programa e endereçam os conceitos fundamentais das redes de comunicações. Os 2 projetos desenvolvidos no laboratório consolidam os conceitos fundamentais, permitem o contacto com as tecnologias de comunicações e focam-se na implementação, configuração e teste de mecanismos e componentes essenciais de uma rede de comunicações.*

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

*The theoretical classes will introduce the syllabus topics and address the fundamental concepts of communication networks. The 2 projects developed in the laboratory consolidate the fundamental concepts, allow contact with communications technologies and focus on the implementation, configuration and testing of mechanisms and essential components of a communications network.*

#### 4.4.9. Bibliografia de consulta/existência obrigatória:

*Tanenbaum, A. S., & Wetherall, D. J. (2011). Computer Networks, 5E. Prentice Hall. ISBN: ISBN-10: 0132126958.*

*Bertsekas, D. P., Gallager, R. G., & Humblet, P. (1992). Data Networks, 2nd Edition, Prentice-Hall International. ISBN: 0-13-200916-1.*

*Kurose, J. F. & Ross, K. W. (2016). Computer Networking - a Top-Down Approach, 7th Edition. Pearson. ISBN-10: 9780133594140.*

**4.4.1.1. Designação da unidade curricular:***Sistemas Operativos***4.4.1.1. Title of curricular unit:***Operating Systems***4.4.1.2. Sigla da área científica em que se insere:***EINFC***4.4.1.3. Duração:***Semestral***4.4.1.4. Horas de trabalho:***162***4.4.1.5. Horas de contacto:***52 (26T + 26TP)***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):***Ricardo Jorge Gomes Lopes da Rocha (T - 26h - 2 turmas; TP - 52h - 11 turmas)***4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:***Jorge Alves da Silva (T - 26h - 2 turmas)**José Manuel de Magalhães Cruz (TP - 26h - 11 turmas)**Luís Miguel Barros Lopes (TP - 52h - 11 turmas)**Pedro Miguel Moreira da Silva (TP - 78h - 11 turmas)**Docente a contratar (TP - 78h - 11 turmas)***4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):***Os objetivos principais desta unidade curricular são fornecer os conhecimentos fundamentais sobre:**O1- a estrutura e o funcionamento de um sistema operativo;**O2- a utilização da interface de programação (API) de um sistema operativo real.**Os estudantes que concluírem com sucesso esta unidade curricular, deverão ser capazes de:**OA1- descrever as funções e a estrutura geral de um sistema operativo, e identificar as abstrações principais que ele fornece;**OA2- descrever o funcionamento dos componentes essenciais de um sistema operativo, a forma como esses componentes interagem entre si e os algoritmos fundamentais usados na sua implementação;**OA3- desenvolver programas que usem e explorem a API de um sistema operativo concreto (Unix/Linux).***4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):***The main objectives of this curricular unit are to provide the fundamental knowledge on:**O1- the structure and the functioning of an operating system;**O2- the use of the application programming interface (API) of a real .**The students who complete successfully this curricular unit must be able:**LO1- to describe the functions and the general structure of an operating system and to identify the main abstractions that it provides;**LO2- to describe the functioning of the essential components of an operating system, the way they interact and the fundamental algorithms used to implement them;**LO3- to develop programs using and exploring the API of a real operating system (Unix/Linux).***4.4.5. Conteúdos programáticos:***1. INTRODUÇÃO AOS SISTEMAS OPERATIVOS**- Objetivos, funções e estrutura geral de um S.O.**- Interface de programação*

**2. GESTÃO DE PROCESSOS**

- Processos e "threads"; sua descrição; estados de um processo
- Estratégias de escalonamento
- Execução concorrente: problemas e soluções
- Mecanismos de sincronização entre processos
- Bloqueio mútuo

**3. GESTÃO DE MEMÓRIA**

- Gestão básica de memória
- Gestão de memória virtual

**4. SISTEMA DE FICHEIROS**

- Interface e implementação do sistema de ficheiros

**5. SISTEMAS DE ENTRADA/SAÍDA**

- Hardware e software de entrada/saída

Tema complementar (coberto ao longo da unidade curricular):

**6. PROGRAMAÇÃO DE SISTEMA (utilização da API do UNIX/LINUX)**

- Manipulação de ficheiros e diretórios.
- Criação e gestão de processos e "threads".
- Intercomunicação entre processos: sinais, "pipes", FIFOs, e memória partilhada.
- Sincronização entre processos e "threads": semáforos, mutexes e variáveis de condição.

**4.4.5. Syllabus:****1. INTRODUCTION TO OPERATING SYSTEMS**

- Objectives, functions and general structure of an operating system
- Programming interface

**2. PROCESS MANAGEMENT**

- Processes and threads; process status
- Scheduling strategies
- Concurrent execution: problems and solutions
- Synchronization mechanisms between processes
- Deadlocks

**3. MEMORY MANAGEMENT**

- Basic memory management
- Virtual memory management

**4. FILE SYSTEMS**

- Interface and implementation of a file system

**5. INPUT/OUTPUT SYSTEMS**

- Input/output hardware and software

**6. SYSTEM PROGRAMMING (API, UNIX/LINUX) - covered along the course**

- File and directory manipulation.
- Process and thread creation and management.
- Interprocess communication: signals, pipes, FIFOs, and shared memory.
- Process and thread synchronization: semaphores, mutexes and condition variables.

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

Os conteúdos programáticos são coerentes com os objetivos da unidade curricular pois cobrem não só os aspetos teóricos do funcionamento dos sistemas operativos – tópicos 1 a 5 do programa – como os aspetos práticos de utilização da interface de programação (API) de um sistema operativo real (Unix/Linux) – tópico 6.

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

The syllabus is consistent with the objectives of the curricular unit because it covers both the theoretical aspects of operating systems functioning - topics 1 to 5 - and the practical aspects of the use of the application programming interface (API) of a real operating system (Unix/Linux) - topic 6.

**4.4.7. Metodologias de ensino (avaliação incluída):**

*Aulas teóricas: exposição e discussão das matérias do programa, com apresentação e encaminhamento da resolução de pequenos exercícios ilustrativos de uso da API de Linux.*

*Aulas teórico-práticas: resolução de exercícios de programação envolvendo a utilização da API.*

*Auto-aprendizagem: estudo das matérias, recorrendo à bibliografia e ao material disponibilizado na página Web da unidade curricular; resolução de exercícios e realização de dois trabalhos práticos, envolvendo a utilização da API.*

*Tipo de Avaliação: distribuída com exame final.*

*Condições para obtenção de frequência: obter 40% em AvD de 30% em MT (ver abaixo).*

*Fórmula de avaliação: avaliação distribuída (AvD) com exame final (AvE).*

- AvD - calculada com base no resultado de um "mini-teste" (MT), a realizar em meados do semestre, e do segundo dos trabalhos práticos propostos (TP2).
- $AvD = MT * 40\% + TP2 * 60\%$

- *Classificação final = AvD\* 50% + AvE \* 50%*
- *É condição de aprovação a obtenção de um mínimo de 40% em AvE.*

#### 4.4.7. Teaching methodologies (including students' assessment):

*Lectures: exposure and discussion of the subject matter, accompanied by the resolution of small illustrative examples of the usage of the Unix/Linux API.*

*Practical lessons: resolution of programming exercises involving the use of the API.*

*Self-learning: study of the program themes, using the bibliography and the materials available at the web page of the curricular unit; resolution of exercises and development of two projects, involving the use of the API.*

*Type of evaluation: distributed with final exam.*

*Terms of frequency: obtain a minimum of 40% in AvD and a minimum of 30% in MT (see below).*

*Formula of evaluation: Distributed evaluation (AvD) with final exam (AvE).*

- *AvD - is based on the result obtained in a short exam (MT) and of the second of the 2nd programming project (TP2).*  
 $AvD = MT * 40\% + TP2 * 60\%$
- *Final grading = AvD \* 50% + AvE \* 50%*
- *To be approved in this course, a minimum of 40% for the AvE evaluation component is required.*

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

*As metodologias de ensino utilizadas permitem que os estudantes tenham a possibilidade de, em sala de aula e em casa, obter as competências teóricas e práticas fundamentais sobre sistema operativos.*

*Os tópicos de 1 a 5 do programa são cobertos nas aulas teóricas. Os tópicos de 6 são cobertos, nas aulas teóricas e teórico-práticas, de forma intercalada com os outros.*

*As aulas teóricas contribuem principalmente para os objetivos de aprendizagem OA1 e OA2 e, parcialmente, para OA3. Nestas aulas são apresentados os conceitos teóricos básicos sobre o funcionamento dos sistemas operativos e são analisadas as principais funcionalidades disponibilizadas pela API de um sistema operativo concreto (Unix/Linux). A apresentação das matérias é acompanhada pela colocação de questões através das quais se procura promover a discussão dos assuntos e pela resolução de pequenos exercícios ilustrativos da utilização da API. Os tópicos de cada um dos dois grandes temas ("Estrutura e funcionamento de um sistema operativo" e "Programação de sistema") são cobertos de uma forma intercalada, o que permite uma melhor compreensão de alguns dos conceitos teóricos através da apresentação de exemplos concretos de aplicação, recorrendo à API de um sistema operativo real.*

*As aulas teórico-práticas e as atividades realizadas fora das aulas contribuem essencialmente para o objetivo OA3 – "desenvolvimento de programas que utilizem e explorem a API de um sistema operativo concreto". Nestas aulas, os estudantes resolvem exercícios de programação focados em cada uma das diferentes funcionalidades da API, à medida que estas vão sendo apresentadas nas aulas teóricas; é fomentada a discussão entre estudantes sobre os problemas que vão surgindo e as propostas de solução que vão sendo apresentadas. Os conhecimentos são consolidados através da realização de dois trabalhos práticos, consistindo no desenvolvimento, por grupos de dois estudantes, de programas que implicam o uso de um conjunto mais alargado de funcionalidades do que os exercícios das aulas e mesmo a exploração de funcionalidades não apresentadas nas aulas.*

*Ficam deste modo cobertos os resultados de aprendizagem pretendidos.*

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

*The used teaching methodologies allow the students to obtain, at the classroom and at home, the fundamental theoretical and practical competences on operating systems. The topics 1 to 5 of the program are covered in the theoretical classes. The topics in 6 are covered both in theoretical and theoretico-practical classes, interleaved with the other topics.*

*The theoretical classes contribute mainly to the learning outcomes LO1 and LO2 and, partially, to LO3. In these classes the basic theoretical concepts about operating systems functioning and the main functionalities available in the API of a real operating system (Unix/Linux) are presented. The presentation is accompanied by questions made to the audience, aiming to promote the discussion of the subject matters, and by the resolution of small exercises illustrating the use of the API. The topics of each one of the two major themes ("Structure and functioning of an operating system" and "System programming" are covered in an interleaved way, thus promoting a better understanding of some of the theoretical concepts through the presentation of concrete application examples, using the API of a real operating system.*

*The theoretical/practical classes and the work done at home contribute mainly to learning objective LO3 - "development of programs that use and explore the API of a real operating system". In these classes, students must solve a set of exercises focused on each of the different functionalities provided by the API, as they are presented in the theoretical classes; the discussion about the arising problems is promoted among students. The knowledge is consolidated through the development of two projects, consisting of the development, by groups of two students, of programs requesting the use of a larger set of functionalities than that used in the exercises and even the search for functionalities that have not been covered in the classes.*

#### 4.4.9. Bibliografia de consulta/existência obrigatória:

*Silberschatz, A., Galvin, P., & Gagne, G. (2013). Operating System Concepts (9th ed). John Wiley & Sons, Inc..*

*Tanenbaum, A., & Bos, H. (2015). Modern Operating Systems. Pearson Education Limited.*

*Stallings, W. (2017). Operating Systems: Internals and Design Principles (9th ed). Pearson.*

*Arpaci-Dusseau, R., & Arpaci-Dusseau, A. (2018). Operating Systems: Three Easy Pieces. Arpaci-Dusseau Books.*

*Robbins, K., & Robbins, S. (2015). UNIX Systems Programming: Communication, Concurrency And Threads (2th ed). Prentice Hall.*

*Kerrisk, M. (2010). The Linux Programming Interface: A Linux and UNIX System Programming Handbook (1st ed). No Starch Press.*  
*Stevens, R., & Rago, S. (2013). Advanced Programming in the UNIX Environment (3rd edition), Addison-Wesley Professional .*  
*Matthew, N., & Stones, R. (2007). Beginning Linux Programming (4th ed). Wrox.*  
*Marques, J., Ferreira, P., Ribeiro, C., Veiga, L., & Rodrigues, R. (2012). Sistemas Operativos (2.a ed). FCA.*

#### Mapa IV - Teoria da Computação

##### 4.4.1.1. Designação da unidade curricular:

*Teoria da Computação*

##### 4.4.1.1. Title of curricular unit:

*Theory of Computation*

##### 4.4.1.2. Sigla da área científica em que se insere:

*EINFC*

##### 4.4.1.3. Duração:

*Semestral*

##### 4.4.1.4. Horas de trabalho:

*162*

##### 4.4.1.5. Horas de contacto:

*52 (26T + 26TP)*

##### 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):

*Nelma Resende Araújo Moreira (T - 26h - 2 turmas; TP - 52h - 12 turmas)*

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

*João Manuel Paiva Cardoso (T - 26h - 2 turmas)*  
*Luís Filipe Guimarães Teófilo (TP - 104h - 12 turmas)*  
*Sabine Babette Broda (TP - 52h - 12 turmas)*  
*Docente a contratar (TP - 104h - 12 turmas)*

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

*O1: Preparar os estudantes em tópicos relacionados com teoria da computação, com um ênfase em linguagens formais.*  
*O2: Munir os estudantes dos conhecimentos que lhes permitam utilizar linguagens regulares e não-regulares, expressões regulares, autómatos finitos deterministas e não-deterministas, linguagens e gramáticas livres de contexto, autómatos de pilha, e máquinas de Turing (TMs).*  
*O3: Capacitar os estudantes para que sejam capazes de expressar problemas computacionais usando linguagens formais, autómatos e TMs.*  
*O4: Capacitar os estudantes de métodos para formalizar problemas computacionais relacionados com linguagens e para provar afirmações relacionadas com esses problemas.*  
*O5: Capacitar os estudantes dos aspetos principais relacionados com linguagens sensíveis ao contexto, recursivas, e recursivamente enumeráveis.*  
*O6: Capacitar os estudantes para relacionarem os modelos de computação estudados com as suas aplicações na teoria da computabilidade e da complexidade.*

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

*O1: To prepare students about computing theory topics with an emphasis to formal language topics.*  
*O2: Students will learn about regular and non-regular languages, regular expressions, deterministic and nondeterministic finite automata, context-free languages and grammars, deterministic and non-deterministic*

*pushdown automata, and Turing machines, and how to apply these topics to problems.*

*O3: Students will be able to express computing problems by using formal languages, automata and Turing machines.*

*O4: Students will learn how to formally specify computing problems related to formal languages and prove related statements.*

*O5: Students will learn the main aspects related to context-sensitive, recursive and recursively Enumerable languages.*

*O6: Students will be able to relate the studied computing models with their applications in the computability theory and complexity theory.*

#### 4.4.5. Conteúdos programáticos:

*P1: Teoria dos Autómatos; Autómatos Finitos;*

*P2: Expressões Regulares e Linguagens Regulares;*

*P3: Propriedades das Linguagens Regulares;*

*P4: Gramáticas e Linguagens sem Contexto;*

*P5: Autómatos de Pilha;*

*P6: Propriedades das Linguagens sem Contexto;*

*P7: Máquinas de Turing;*

*P8: Complexidade Computacional dos Algoritmos Principais;*

*P9: Introdução às Linguagens Sensíveis ao Contexto e às Linguagens Recursivas e Recursivamente Enumeráveis;*

*P10: Introdução aos Problemas Tratáveis, Intratáveis e Indecidíveis.*

#### 4.4.5. Syllabus:

*P1: Automata Theory; Finite Automata;*

*P2: Regular Expressions and Languages;*

*P3: Properties of Regular Languages;*

*P4: Context-Free Grammars and Languages;*

*P5: Pushdown Automata;*

*P6: Properties of Context-Free Language;*

*P7: Turing Machines;*

*P8: Computational Complexity of the Main Algorithms;*

*P9: Introduction to Context-sensitive, Recursive and Recursively Enumerable Languages;*

*P10: Introduction to Tractable, Intractable, and Undecidable Problems.*

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

*O elenco programático apresentado é essencial para os objetivos de aprendizagem propostos e para a aquisição das competências descritas. Os conteúdos programáticos cobrem diretamente os aspetos teóricos relevantes para os objetivos de aprendizagem. A correspondência entre módulos do programa e objetivos de aprendizagem é a seguinte:*

*P1 -> O1, O2*

*P2 -> O1, O2*

*P3 -> O1, O2, O4*

*P4 -> O1, O2*

*P5 -> O2*

*P6 -> O2, O4*

*P7 -> O3*

*P8 -> O6*

*P9 -> O5*

*P10 -> O6*

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

*The content presented in the syllabus is essential to the learning outcomes proposed and to the acquisition of the competences described. The syllabus covers theoretical aspects directly relevant to the learning outcomes. The correspondence between program modules and learning outcomes is as follows:*

*(For each module, we indicate the more directly targeted outcomes.)*

*P1 -> O1, O2*

*P2 -> O1, O2*

*P3 -> O1, O2, O4*

*P4 -> O1, O2*

*P5 -> O2*

*P6 -> O2, O4*

*P7 -> O3*

*P8 -> O6*

*P9 -> O5*

*P10 -> O6*

#### 4.4.7. Metodologias de ensino (avaliação incluída):

*As aulas teóricas são usadas para exposição formal da matéria, acompanhada da apresentação de exemplos, realização de exercícios e sua discussão.*

*Nas aulas teórico-práticas são propostos exercícios de aplicação.*

*São realizados dois minitestes, sensivelmente a seis e doze semanas do início do semestre, com o objetivo de testar se os conceitos básicos estão a ser dominados pela generalidade dos alunos.*

*O esforço previsto para além das aulas é de cerca de 4h semanais.*

*Tipo de Avaliação: Avaliação distribuída com exame final*

*Condições de Frequência: Avaliação distribuída (AD) não inferior a 7 valores, com nota de cada miniteste (MT) não inferior a 6 valores, e um máximo de 3 faltas não justificadas (25%) nas aulas TP.*

*Fórmula de avaliação: AD: Avaliação Distribuída constituída pelas componentes MT1 e MT2 = 0,5 MT1 + 0,5 MT2 (mín: 7 valores)*

*MT1 e MT2: minitests 1 e 2, respetivamente (mínimo: 6 valores em cada MT)*

*EF: exame final (mín: 7 valores)*

*Nota=arredonda(0,4 AD + 0,6 EF).*

#### 4.4.7. Teaching methodologies (including students' assessment):

*In theoretical classes the contents are formally exposed along with presentation and discussion of examples.*

*In tutorial classes application exercises are proposed.*

*Two mini-test will be held, approximately six and twelve weeks from the start of the semester, to check if the basic concepts are being understood by the majority of students.*

*The foreseen effort beyond classes is of about 4h per week.*

*Type of evaluation: Distributed evaluation with final exam*

*Terms of frequency: Distributed evaluation not inferior to 7 marks, with the grade of each mini-test (MT) not inferior to 6 marks, and a maximum of 3 non-justified absences (25%) on the tutorial classes.*

*Formula of evaluation: AD: Distributed Evaluation consists of two components, MT1 and MT2 = 0.5 MT1 + 0.5 MT2 (min: 7 marks)*

*MT1 and MT2: mini-tests 1 and 2, respectively (min: 6 marks for each)*

*EF: final exam (min: 7 marks)*

*Grade = rounded(0.4 AD + 0.6 EF).*

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

*Em complemento a componentes de exposição de exemplos e de casos de estudo, uma componente importante para a aquisição de parte do conhecimento é a realização de exercícios por parte dos estudantes.*

*A exposição de tópicos e a aplicação dos mesmos a problemas, incluindo a prova de enunciados, contribui definitivamente para que os estudantes fiquem familiarizados com os assuntos relacionados com teoria da computação.*

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

*Besides the exposition component of examples and use cases in this course, an important component for the acquisition of knowledge is through the resolution of exercises by students.*

*The exposure of the topics and their application to problems, including the proofs of statements, contribute decisively to enable students to familiarize themselves with the body of knowledge related to theory of computation.*

#### 4.4.9. Bibliografia de consulta/existência obrigatória:

*Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Introduction to Automata Theory, Languages, and Computation (3rd edition). Addison Wesley. ISBN-13: 978-0321455369.*

*Sipser, M. (2013). Introduction to the Theory of Computation (3rd edition). Cengage Learning. ISBN: 978-1-133-18779-0.*

*Sudkamp, T. A. (2005). Languages and Machines: an Introduction to the Theory of Computer Science (3rd edition). Pearson Education. ISBN-13: 978-0321322210*

### 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 perfil de formação que este ciclo de estudos pretende atingir conduzirá a graduados com uma formação cuidada nas ciências básicas de Engenharia e uma formação abrangente nos fundamentos das grandes áreas científicas e técnicas da Engenharia Informática. Tendo em conta que o percurso do estudante ao longo do ciclo de estudos vai evoluindo de UCs com formação das ciências básicas até às de âmbito mais profissionais, as metodologias vão-se também adaptando, da formação mais teórica no 1º e 2º ano, até à formação de natureza mais aplicada, com maior autonomia dos estudantes, no 2º e 3º ano, nomeadamente nos Laboratórios e no Projeto Integrador. As tipologias de aulas (T, TP, PL, OT) e metodologias de ensino (resolução de problemas, ensino baseado em projeto, etc.) de cada UC estão adaptadas ao nível de aprendizagem pretendido (conhecimento, compreensão, aplicação, análise, etc.).*

#### 4.5.1. Evidence of the teaching and learning methodologies coherence with the intended learning outcomes of the study programme:

*The education profile that this course of study aims to achieve will lead to graduates with a thorough education in basic engineering sciences and a comprehensive education in the foundations of the major scientific and technical areas of informatics engineering. Bearing in mind that the student's journey along the study cycle will evolve from UCs with an education in basic engineering sciences to UCs with a more applied scope, the methodologies also evolve from a more theoretical education in the 1st and 2nd year to a more applied education, with greater autonomy of students, in the 2nd and 3rd year, namely in Laboratories and the Capstone Project. The class typologies (T, TP, PL, OT) and teaching methodologies (problem solving, project-based teaching, etc.) of each UC are adapted to the intended learning level (knowledge, comprehension, application, analysis, etc.).*

#### **4.5.2. Forma de verificação de que a carga média de trabalho que será necessária aos estudantes corresponde ao estimado em ECTS:**

*No início de cada ano letivo os docentes planificam na ficha da UC as atividades da UC (aulas, trabalhos, estudo individual, provas de avaliação, etc.), incluindo a calendarização e a estimativa de carga de trabalho de cada atividade (por forma a totalizar o nº de ECTS x 27h), procurando distribuir uniformemente e evitar picos de trabalho no tempo. As fichas das UC são validadas pelo Diretor do ciclo de estudos. Relativamente aos trabalhos, onde podem existir maiores variações, diversas UCs solicitam aos estudantes que reportem o tempo despendido na sua realização, permitindo assim tomar medidas corretivas. No final do semestre, os docentes responsáveis por cada UC elaboram um relatório onde comentam o seu funcionamento e a sua adequação ao trabalho exigido. É ainda efetuado um inquérito aos estudantes, via SIGARRA, onde podem exprimir as suas opiniões sobre o trabalho exigido pelas UCs. Os estudantes são também auscultados através da Comissão de Acompanhamento e representantes de ano.*

#### **4.5.2. Means to verify that the required students' average workload corresponds the estimated in ECTS.:**

*At the beginning of each school year, the teachers plan in the UC form the activities of the UC (classes, assignments, study, evaluation, etc.), including the schedule and estimated workload of each activity (in order to total the number of ECTS x 27h), trying to distribute evenly the workload and avoid spikes in time. The UC forms are validated by the course Director. With regard to assignments, where there may exist greater variations, several UCs ask students to report the time spent on them, thus allowing corrective action to be taken. At the end of the semester, the teachers responsible for each UC prepare a report commenting on its operation and its adequacy to the required work. Students are also surveyed via SIGARRA, where they can express their opinions about the work required by the UCs. Students are also listened to through the Monitoring Committee and year representatives.*

#### **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:**

*Os métodos, componentes, pesos e datas de avaliação, tal com os objetivos de aprendizagem, são definidos na ficha de cada UC pelos docentes responsáveis, sendo validados pelo Diretor do ciclo de estudos. De acordo com os princípios de Bolonha, o método de avaliação das UCs evoluiu de um método baseado principalmente em exame final para um método predominante misto, combinando o exame final com uma componente de avaliação distribuída com peso significativo, ou mesmo baseado unicamente em avaliação distribuída, como acontece em todas as UCs de Laboratório. As UCs de ensino de programação incorporam normalmente componentes de avaliação em computador. As competências de comunicação são avaliadas através de relatórios e apresentações. O trabalho desenvolvido dentro e fora das aulas, devidamente acompanhado e avaliado pelos docentes, integra-se nas próprias metodologias de aprendizagem baseadas em projeto.*

#### **4.5.3. Means of ensuring that the students assessment methodologies are adequate to the intended learning outcomes:**

*The methods, components, weights and dates of the assessments, together with the learning outcomes, are defined in the form of each UC by the responsible teachers, being validated by the Director of the study cycle. In accordance with the Bologna principles, the method of evaluation of the UCs has evolved from a method based mainly on the final exam to a predominantly mixed method, combining the final exam with a significant weighted distributed evaluation component or even based solely on distributed evaluation, as happens with all Lab UCs. Programming UCs typically incorporate computer assessment components. Communication skills are assessed through reports and presentations. The work developed inside and outside the classes, properly monitored and evaluated by the teachers, is an integral part of the project-based learning methodology.*

#### **4.5.4. Metodologias de ensino previstas com vista a facilitar a participação dos estudantes em atividades científicas (quando aplicável):**

- O interesse e envolvimento dos estudantes em atividades de investigação científica é promovido e facilitado em 1º lugar por via do corpo docente, na sua maioria integrado em centros de I&D;
- Em diversas UCs, os estudantes são confrontados com resultados de investigação na fronteira do conhecimento, que são desafiados a analisar criticamente;
- Em algumas UCs, os estudantes são desafiados a encontrar soluções inovadoras para problemas reais e a documentá-las na forma de artigo científico, tendo já alguns trabalhos originado publicações científicas;
- Algumas UCs integram por vezes experiências científicas com a participação dos estudantes;
- Na UC de Projeto Integrador, uma das modalidades previstas é o desenvolvimento de um projeto de iniciação à investigação numa unidade de I&D sob tutoria de um docente;
- Ao longo do ciclo de estudos, diversos estudantes participam em projetos de iniciação à investigação em centros ou laboratório de I&D, com direito a menção no suplemento ao diploma

#### **4.5.4. Teaching methodologies that promote the participation of students in scientific activities (as applicable):**

- *Students' interest and involvement in scientific research activities is promoted and facilitated in first place by the teachers, the majority of which are integrated in R&D centers;*
- *In many UCs, students are confronted with research results at the knowledge frontier, which they are challenged to critically analyze;*
- *In some UCs, students are challenged to find innovative solutions to real problems and to document them in the form of a scientific paper, with some works already leading to scientific publications;*
- *Some UCs may integrate scientific experiments with students' participation;*
- *In the Capstone Project, one of the modalities foreseen is the development of a research initiation project in an R&D unit under the supervision of a teacher;*
- *Throughout the study cycle, several students participate in research initiation projects in R&D centers or laboratories, with the possibility of a mention in the supplement to their diploma (as extracurricular projects).*

#### 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:**

*A estrutura curricular da LiEIC cumpre os requisitos definidos no artigo 9.º do Decreto-Lei n.º 74/2006, na redação dada pelo Decreto-Lei n.º 65/2018, de 16 de agosto, para ciclo de estudos conducente ao grau de licenciado no ensino universitário. De facto, a LiEIC está configurada como um ciclo de estudos com a duração de 6 semestres curriculares correspondentes a um total de 180 créditos ECTS, cumprindo assim o estipulado nesse artigo (que admite um intervalo de 180 a 240 créditos ECTS e seis a oito semestres curriculares). A opção por uma duração de 6 semestres está alinhada com a duração típica de 1ºs ciclos na área de Engenharia Informática no espaço europeu, com a prática comum das Escolas de Engenharia em Portugal, e com a duração do 1º ciclo do anterior ciclo de estudos integrado (MIEIC) a que a LiEIC sucede em cumprimento do artigo 19.º do Decreto-Lei n.º 65/2018 de 16 de agosto.*

**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:**

*The curricular structure of LiEIC meets the requirements set out in Article 9 of Decree-Law No. 74/2006, in the wording given by Decree-Law n.º 65/2018, of August 16, for a study cycle leading to a university degree of "licenciado". In fact, LiEIC is configured as a course of study lasting 6 curricular semesters corresponding to a total of 180 ECTS credits, thus fulfilling the stipulated in that article (which admits a range of 180 to 240 ECTS credits and six to eight semesters). The option for a duration of 6 semesters is aligned with the typical duration of 1st cycles in the area of Informatics Engineering in the European area, with the common practice of Engineering Schools in Portugal, and with the duration of the 1st cycle of the previous integrated study cycle (MIEIC) to which LiEIC succeeds in compliance with Article 19 of Decree-Law No. 65/2018 of 16 August.*

**4.6.2. Forma como os docentes foram consultados sobre a metodologia de cálculo do número de créditos ECTS das unidades curriculares:**

*O nº de créditos ECTS por UC é harmonizado a nível da UP para facilitar a mobilidade interna, a partilha de UCs entre ciclos de estudo e o alinhamento com as boas práticas nacionais e internacionais. Assim, as UCs da LiEIC foram concebidas com um âmbito e volume de trabalho correspondente em geral a 6 créditos ECTS, existindo algumas UCs de 4,5 ECTS e UCs relacionadas com Desenvolvimento Pessoal e Competências Transversais dimensionadas com 1,5 créditos ECTS.*

*Após uma primeira definição do mapa de UCs e seus objetivos gerais e créditos ECTS pela Comissão Científica (CC), seguiu-se um período de auscultação do corpo docente. Com base no feedback recebido, o mapa inicial foi refinado e sujeito a apreciação pelos Conselhos dos Departamentos envolvidos (DEI e DCC).*

*De seguida, a CC identificou regentes potenciais de cada UC, a quem solicitou a elaboração da respetiva ficha com base nos objetivos gerais definidos, com o cuidado de adequar o âmbito e volume de trabalho ao nº de créditos ECTS*

**4.6.2. Process used to consult the teaching staff about the methodology for calculating the number of ECTS credits of the curricular units:**

*The number of ECTS credits per UC is harmonized at UP to facilitate internal mobility, the sharing of UCs, and the alignment with national and international practice. Hence, the LiEIC UCs were designed with a scope and workload generally corresponding to 6 ECTS credits, with some UCs with 4.5 ECTS. There are also small UCs with 1.5 credits related with Personal Development and Transferable Skills.*

*Following a first definition of the map of UCs and their general objectives and ECTS credits by the Scientific Commission (CC), a period of faculty listening followed. Based on the feedback received, the initial map was refined and subject to review by the councils of the main Department Councils involved (DEI and DCC).*

*Subsequently, the CC identified potential regents for each UC, to whom it requested the preparation of the respective form based on the general objectives defined, taking care to adjust the scope and workload to the number of ECTS credits*

#### 4.7. Observações

---

**4.7. Observações:**

*O ciclo de estudos está estruturado em UCs que totalizam 180 créditos ECTS, dos quais 175.5 em UCs obrigatórias e 4.5 em UCs optativas relacionadas com Competências Transversais, culminando num Projeto Integrador que decorre no último semestre, em paralelo com outras UCs.*

*A aprovação em todas as unidades curriculares do plano de estudos permitirá a obtenção do grau de licenciado em Engenharia Informática e Computação.*

*Observações do Mapa III:*

*(a) A UC “Projeto UP” é focada no desenvolvimento de competências pessoais e interpessoais fundamentais para a integração no ensino superior.*

*(b) Nas UCs de Competências Transversais I, II e III os estudantes devem escolher livremente, de entre um catálogo a oferecer pela UP em cada semestre, as unidades que pretendem realizar. A participação em concursos de programação, na organização de eventos ou em estágios de verão também é elegível, mediante autorização do diretor de ciclo de estudos e desde que essas atividades estejam acreditadas e creditadas na UP. O catálogo a oferecer incluirá unidades relacionadas com aspetos sociais e profissionais recomendados como obrigatórios para 1<sup>os</sup> ciclos no Computer Science Curricula 2013 do ACM e IEEE Computer Society, nomeadamente sobre comunicação profissional, ética profissional, propriedade intelectual e privacidade e liberdades civis. As escolhas dos estudantes são sujeitas a validação pelo diretor do ciclo de estudos, podendo ser condicionada à realização de pelo menos uma unidade relacionada com o referido referencial (ficando 2 unidades de escolha livre e 1 de escolha condicionada).*

*(c) Na UC de Projeto Integrador, os estudantes, organizados em equipas de média dimensão, realizam um projeto de Engenharia Informática com impacto social, para resolução de um problema real, com base em propostas oriundas preferencialmente de organizações externas, culminando num evento de exposição pública dos resultados alcançados. Em alternativa, sob proposta do estudante, e mediante autorização do diretor do ciclo de estudos, os estudantes poderão realizar um estágio em ambiente empresarial ou numa unidade de I&D (para iniciação à investigação), sob tutoria de um docente, ou um projeto multidisciplinar, orientado por um docente da FEUP ou FCUP.*

*(d) A partir do 3<sup>o</sup> semestre, o plano de estudos contempla uma sequência de UCs laboratoriais, para aplicação e aprofundamento dos conhecimentos adquiridos e desenvolvimento de competências pessoais e interpessoais.*

#### **4.7. Observations:**

*The study cycle is structured in UCs totaling 180 ECTS credits, of which 175.5 in compulsory UCs and 4.5 in optional UCs related to Transversal Skills, culminating in a Capstone Project that takes place in the last semester, in parallel with other UCs.*

*Approval in all curriculum units of the syllabus gives a “licenciatura” degree in Informatics Engineering and Computing.*

*Notes on Map III:*

*(a) The “Project UP” curricular unit is focused on the development of personal and interpersonal competencies essential for integration in higher education.*

*(b) In the UCs on Transferable Skills I, II and III, students are free to choose from a catalog to be offered by UP which units they intend to undertake. Participation in programming contests, organization of events or summer internships is also eligible, subject to authorization by the study cycle director and provided that these activities are accredited by UP. The catalog to be offered will include units related to social and professional aspects recommended as compulsory for 1st cycles in the Computer Science Curriculum 2013 of the ACM and IEEE Computer Society, namely on professional communication, professional ethics, intellectual property, privacy and civil liberties. Students' choices are subject to validation by the study cycle director, and may be conditioned to the completion of at least one unit related to the above referential (resulting in 2 units of free choice and 1 of conditioned choice).*

*(c) In the Capstone Project, the students, organized in medium-sized teams, undertake an Informatics Engineering project with social impact, to solve a real problem, based on proposals preferably from outside organizations, culminating in a seminar event for public exposure of the results achieved. Alternatively, upon the student's proposal, and upon the authorization of the study cycle director, students may undertake an internship in a business setting or in an R&D unit (for research initiation), under the guidance of a teacher, or a multidisciplinary project, supervised by a FEUP ou FCUP teacher.*

*(d) Starting from the 3rd semester, the syllabus includes a sequence of laboratory UCs, for application and deepening of acquired knowledge and development of personal and interpersonal skills.*

## **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.**

*Direção do ciclo de estudos:*

- João Carlos Pascoal Faria (Codiretor, Prof. Associado na FEUP a tempo inteiro)
- Luís Miguel Barros Lopes (Codiretor, Prof. Associado na FCUP a tempo inteiro)
- Maria Cristina de Carvalho Alves Ribeiro (Diretor Adjunto, Prof. Associado na FEUP a tempo inteiro)

*Comissão Científica:*

- João Carlos Pascoal Faria (Prof. Associado na FEUP a tempo inteiro)
- Jorge Manuel Gomes Barbosa (Prof. Auxiliar na FEUP a tempo inteiro)
- Luís Miguel Barros Lopes (Prof. Associado na FCUP a tempo inteiro)
- Luís Paulo Gonçalves dos Reis (Prof. Associado na FEUP a tempo inteiro)
- Maria Cristina de Carvalho Alves Ribeiro (Prof. Associada na FEUP a tempo inteiro)
- Pedro Baltazar Vasconcelos (Prof. Associado na FCUP a tempo inteiro)
- Rui Pedro de Magalhães Claro Prior (Prof. Auxiliar na FCUP a tempo inteiro)

### 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
Ademar Manuel Teixeira de Aguiar	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Engenharia de Software)	100	<a href="#">Ficha submetida</a>
Alexandre Miguel Prior Afonso	Professor Auxiliar ou equivalente	Doutor	Engenharia Química	100	<a href="#">Ficha submetida</a>
Alexandre Miguel Barbosa Valle de Carvalho	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	<a href="#">Ficha submetida</a>
Alípio Mário Guedes Jorge	Professor Associado ou equivalente	Doutor	Ciência de Computadores	100	<a href="#">Ficha submetida</a>
Álvaro Pedro de Barros Borges Reis Figueira	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	<a href="#">Ficha submetida</a>
Ana Cristina Costa Aguiar	Professor Auxiliar ou equivalente	Doutor	Redes de Telecomunicações	100	<a href="#">Ficha submetida</a>
Ana Cristina Ramada Paiva	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Engenharia de Software)	100	<a href="#">Ficha submetida</a>
Ana Maria Azevedo Neves	Professor Auxiliar ou equivalente	Doutor	Engenharia Mecânica	60	<a href="#">Ficha submetida</a>
Ana Paula Cunha da Rocha	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Inteligência Artificial)	100	<a href="#">Ficha submetida</a>
Ana Paula da Silva Dias	Professor Auxiliar ou equivalente	Doutor	Matemática	100	<a href="#">Ficha submetida</a>
Ana Paula Nunes Gomes Tomás	Professor Associado ou equivalente	Doutor	Ciência de Computadores	100	<a href="#">Ficha submetida</a>
André Monteiro de Oliveira Restivo	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	<a href="#">Ficha submetida</a>
António Augusto de Sousa	Professor Associado ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Computação Gráfica)	100	<a href="#">Ficha submetida</a>
António Fernando Vasconcelos Cunha Castro Coelho	Professor Associado ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Computação Gráfica)	100	<a href="#">Ficha submetida</a>
António José dos Santos Silva	Professor Auxiliar ou equivalente	Doutor	Física	100	<a href="#">Ficha submetida</a>
António José Duarte Araújo	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Arquitetura de Computadores)	100	<a href="#">Ficha submetida</a>
António Manuel Lucas Soares	Professor Associado ou equivalente	Doutor	Engenharia Eletrotécnica e Computadores (Sistemas de Informação)	100	<a href="#">Ficha submetida</a>
António Mário da Silva Marcos Florido	Professor Associado ou equivalente	Doutor	Ciência de Computadores	100	<a href="#">Ficha submetida</a>
António Miguel da Fonseca Fernandes Gomes	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores	100	<a href="#">Ficha submetida</a>
António Miguel Pontes Pimenta Monteiro	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Processamento de Imagem)	100	<a href="#">Ficha submetida</a>
Ariel Ricardo Negrão da Silva Guerreiro	Professor Auxiliar ou equivalente	Doutor	Engenharia Física	100	<a href="#">Ficha submetida</a>
Armando Luis Ferreira Leitão	Professor Auxiliar ou equivalente	Doutor	Engenharia de Produção e Gestão	100	<a href="#">Ficha submetida</a>
Augusto da Silveira Rodrigues	Professor Auxiliar ou equivalente	Doutor	Física	100	<a href="#">Ficha submetida</a>
Bruno Miguel Carvalhido Lima	Assistente convidado ou equivalente	Mestre	Engenharia Informática e Computação	28.6	<a href="#">Ficha submetida</a>
Bruno Serra Loff Barreto	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	<a href="#">Ficha submetida</a>
Carla Alexandra Teixeira Lopes	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	<a href="#">Ficha submetida</a>
Carla Susana Santana Carmelo Rosa	Professor Auxiliar ou equivalente	Doutor	Física	100	<a href="#">Ficha submetida</a>
Carlos Manuel Milheiro de Oliveira Pinto Soares	Professor Associado ou	Doutor	Ciência de Computadores	100	<a href="#">Ficha submetida</a>

	equivalente					
Catarina Rosa Santos Ferreira de Castro	Professor Auxiliar ou equivalente	Doutor	Ciências da Engenharia	100	Ficha submetida	
Daniel Augusto Gama de Castro Silva	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	Ficha submetida	
Eduardo Resende Brandão Marques	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida	
Fernando José Cassola Marques	Assistente ou equivalente	Mestre	Tecnologias de Informação e Comunicação na Educação	25	Ficha submetida	
Fernando Manuel Augusto da Silva	Professor Catedrático ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida	
Filipe Alexandre Pais de Figueiredo Correia	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	Ficha submetida	
Francisco Manuel Madureira e Castro Vasques de Carvalho	Professor Associado ou equivalente	Doutor	Informática Industrial	100	Ficha submetida	
Gabriel de Sousa Torcato David	Professor Associado ou equivalente	Doutor	Informática	100	Ficha submetida	
Gil Coutinho Costa Seixas Lopes	Assistente convidado ou equivalente	Mestre	Redes e Serviços de Comunicação	25	Ficha submetida	
Gil Manuel Magalhães de Andrade Gonçalves	Professor Auxiliar ou equivalente	Doutor	Engenharia Electrotécnica e Computadores (Informática Industrial)	100	Ficha submetida	
Gilberto Bernardes de Almeida	Professor Auxiliar ou equivalente	Doutor	Media Digitais	100	Ficha submetida	
Henrique Daniel de Avelar Lopes Cardoso	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	Ficha submetida	
Hugo José Sereno Lopes Ferreira	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	Ficha submetida	
Inês de Castro Dutra	Professor Auxiliar ou equivalente	Doutor	Ciências de Computadores	100	Ficha submetida	
Jaime Enrique Villate Matiz	Professor Auxiliar ou equivalente	Doutor	Física	100	Ficha submetida	
Joana Cassilda Rodrigues Espain de Oliveira	Professor Auxiliar convidado ou equivalente	Doutor	Física	100	Ficha submetida	
João António Correia Lopes	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida	
João Carlos Pascoal Faria	Professor Associado ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Engenharia de Software)	100	Ficha submetida	
João Manuel Paiva Cardoso	Professor Catedrático ou equivalente	Doutor	Computer Engineering	100	Ficha submetida	
João Miguel Rocha da Silva	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	35.7	Ficha submetida	
João Paulo da Silva Machado Garcia Vilela	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida	
João Paulo de Castro Canas Ferreira	Professor Associado ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Engenharia de Computadores)	100	Ficha submetida	
João Paulo Filipe de Sousa	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores	100	Ficha submetida	
João Pedro Carvalho Leal Mendes Moreira	Professor Auxiliar ou equivalente	Doutor	Ciências de engenharia (Ciência de dados)	100	Ficha submetida	
João Pedro Pedroso Ramos dos Santos	Professor Auxiliar ou equivalente	Doutor	Matemática computacional	100	Ficha submetida	
Jorge Alves da Silva	Professor Auxiliar ou equivalente	Doutor	Engenharia Electrotécnica e de Computadores (Processamento de Imagem)	100	Ficha submetida	
Jorge Manuel Gomes Barbosa	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Sistemas Paralelos)	100	Ficha submetida	
Jorge Miguel Milhazes de Freitas	Professor Associado ou equivalente	Doutor	Matemática	100	Ficha submetida	
José Augusto Trigo Barbosa	Professor Auxiliar ou equivalente	Doutor	Engenharia Mecânica	100	Ficha submetida	
José Luís Cabral Moura Borges	Professor Associado ou equivalente	Doutor	Computer Science	100	Ficha submetida	
José Manuel de Magalhães Cruz	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Sistemas Distribuídos)	100	Ficha submetida	

José Maria Corte Real da Costa Pereira	Professor Auxiliar convidado ou equivalente	Doutor	Electrical and Computer Engineering – Intelligent Systems, Robotics, and Control	28.6	Ficha submetida
José Paulo de Vilhena Geraldes Leal	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
José Ferreira Alves	Professor Catedrático ou equivalente	Doutor	Matemática	100	Ficha submetida
Luís Filipe Guimarães Teófilo	Professor Auxiliar convidado ou equivalente	Doutor	Engenharia Informática	42.9	Ficha submetida
Luís Filipe Coelho Antunes	Professor Catedrático ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Luis Filipe Pinto de Almeida Teixeira	Professor Auxiliar ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Visão por Computador)	100	Ficha submetida
Luís Miguel Barros Lopes	Professor Associado ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Luís Miguel Fortuna Rodrigues Martelo	Professor Auxiliar ou equivalente	Doutor	Física	100	Ficha submetida
Luís Paulo Gonçalves dos Reis	Professor Associado ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Inteligência Artificial)	100	Ficha submetida
Manuel Alberto Pereira Ricardo	Professor Catedrático ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Redes)	100	Ficha submetida
Manuel António Salgueiro da Silva	Professor Auxiliar ou equivalente	Doutor	Física	100	Ficha submetida
Manuel Bernardo Martins Barbosa	Professor Auxiliar ou equivalente	Doutor	Electrical and Electronic Engineering	100	Ficha submetida
Manuel Eduardo Carvalho Duarte Correia	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores/Computer Science	100	Ficha submetida
Manuel Firmino da Silva Torres	Professor Auxiliar ou equivalente	Doutor	Ciências da Educação	100	Ficha submetida
Maria Cristina de Carvalho Alves Ribeiro	Professor Associado ou equivalente	Doutor	Informática/Inteligência Artificial	100	Ficha submetida
Maria Helena Pinto da Rocha Mena de Matos	Professor Auxiliar ou equivalente	Doutor	Matemática	100	Ficha submetida
Maria José Fernandes Vaz Lourenço Marques	Professor Auxiliar ou equivalente	Doutor	Engenharia Mecânica, Especialidade Ciência dos Materiais	100	Ficha submetida
Maria Luísa Romariz Madureira	Professor Auxiliar ou equivalente	Doutor	Engenharia Mecânica	100	Ficha submetida
Maria Teresa Galvão Dias	Professor Auxiliar ou equivalente	Doutor	Ciências da Engenharia	100	Ficha submetida
Maria Teresa Magalhães da Silva Pinto de Andrade	Professor Auxiliar ou equivalente	Doutor	Telecomunicações (Multimedia)	100	Ficha submetida
Michel Celestino Paiva Ferreira	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Miguel Tavares Coimbra	Professor Auxiliar ou equivalente	Doutor	Electronic Engineering	100	Ficha submetida
Nelma Resende Araújo Moreira	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Nuno Honório Rodrigues Flores	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	Ficha submetida
Pedro Alexandre Guimarães Lobo Ferreira Souto	Professor Auxiliar ou equivalente	Doutor	Computer Science	100	Ficha submetida
Pedro Baltazar Vasconcelos	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Pedro Gabriel Dias Ferreira	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Pedro Manuel Pinto Ribeiro	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Pedro Miguel Alves Brandão	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Pedro Miguel Moreira da Silva	Professor Auxiliar convidado ou equivalente	Doutor	Redes e Tecnologias de Informação	17.9	Ficha submetida
Ricardo Jorge Gomes Lopes da Rocha	Professor Associado ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Rita Paula Almeida Ribeiro	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida

Rogério Ventura Lages dos Santos Reis	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Rolando da Silva Martins	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Rosaldo José Fernandes Rossetti	Professor Auxiliar ou equivalente	Doutor	Ciência da Computação	100	Ficha submetida
Rui Carlos Camacho de Sousa Ferreira da Silva	Professor Associado ou equivalente	Doutor	Engenharia Eletrotécnica e de Computadores (Inteligência Artificial)	100	Ficha submetida
Rui Pedro Amaral Rodrigues	Professor Auxiliar ou equivalente	Doutor	Informática	100	Ficha submetida
Rui Pedro de Magalhães Claro Prior	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Sabine Babette Broda	Professor Associado ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Sandra Maria Mendes Alves	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Sérgio Armindo Lopes Crisóstomo	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
Sérgio Sobral Nunes	Professor Auxiliar ou equivalente	Doutor	Engenharia Informática	100	Ficha submetida
Sónia Isabel Silva Pinto	Professor Auxiliar convidado ou equivalente	Doutor	Engenharia Química e Biológica	60	Ficha submetida
Teresa Carla de Canha e Matos	Assistente convidado ou equivalente	Mestre	Engenharia Informática e Computação	25	Ficha submetida
Tiago Boldt Sousa	Assistente convidado ou equivalente	Mestre	Engenharia Informática e Computação	59.3	Ficha submetida
Verónica Costa Teixeira Pinto Orvalho	Professor Auxiliar ou equivalente	Doutor	Computação Gráfica	100	Ficha submetida
Vítor Manuel de Morais Santos Costa	Professor Auxiliar ou equivalente	Doutor	Ciência de Computadores	100	Ficha submetida
				<b>9908</b>	

<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.

106

###### 5.4.1.2. Número total de ETI.

99.04

##### 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:	95	95.92084006462

##### 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):	97.42	98.364297253635

#### 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	74.25	74.969709208401 99.04
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	1.63	1.6457996768982 99.04

#### 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	94	94.911147011309 99.04
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	1.13	1.1409531502423 99.04

#### 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.

*A avaliação do desempenho do pessoal docente seguirá as regras estabelecidas pela legislação aplicável, nomeadamente, o ECDU, o RJIES e o Regulamento de Avaliação de Desempenho da UP. Os Docentes do ciclo de estudos são periodicamente avaliados de acordo com o regime de avaliação de desempenho que consta no Regulamento de avaliação de desempenho dos docentes da Universidade do Porto (Despacho n.º 5880/2017 de 4 de julho) e mais especificamente no Regulamento de Avaliação de Desempenho dos Docentes da Faculdade de Engenharia da Universidade do Porto (Despacho n.º 5096/2012 de 12 de abril de 2012, publicado em Diário da República, 2.ª série — N.º 73) e no Regulamento de Avaliação de Desempenho dos Docentes da Faculdade de Ciências da Universidade do Porto (Despacho n.º 165/2013 de 13 de maio de 2013, publicado no Diário da República, 2.ª série — N.º 91), os quais se encontram em reformulação.*

##### 5.5. Procedures for the assessment of the teaching staff performance and measures for their permanent updating and professional development.

*The performance evaluation of teaching staff will follow the rules established by the applicable legislation, namely the ECDU, RJIES and the UP Performance Evaluation Regulation. Teachers of the study cycle are periodically evaluated in accordance with the performance evaluation regime set out in the Faculty Performance Evaluation Regulation of the University of Porto (Order No. 5880/2017 of July 4), and more specifically in the Performance Evaluation Regulation of the Faculty of Engineering of the University of Porto (Order No. 5096/2012 of April 12, 2012, published in Diário da República, 2nd Series - No. 73) and the Performance Evaluation Regulation of the Faculty of Sciences of the University of Porto (Order No. 165/2013 of May 13, 2012, published in Diário da República, 2nd Series - No. 91), which are currently being reformulated.*

##### 5.6. Observações:

- No quadro 5.3, indicam-se os docentes integrados atualmente na carreira docente ou de investigação, bem como alguns docentes convidados com contrato, que se prevê estarem envolvidos na lecionação do ciclo de estudos, num cenário em que o volume de estudantes é similar ao 1º ciclo (1º a 3º ano) dos atuais ciclos de estudos integrado MIEIC e MIERSI (somados), com 2 turmas teóricas e 9 a 13 turmas práticas por UC. O regime de tempo refere-se ao vínculo com a UP, e não à proporção de dedicação a este ciclo de estudos em particular; de facto, na generalidade dos casos, os docentes colaboram em múltiplos ciclos de estudos.*
- Os docentes alocados para lecionar UCs classificadas na área fundamental do ciclo de estudos (EIC) são especializados em EIC. Os docentes alocados para lecionar UCs nas áreas de Matemática, Física e Competências Transversais não são especializados em EIC (são especializados nas áreas dessas UCs).*
- Nas fichas de algumas unidades curriculares há horas de serviço docente alocadas a docentes a contratar/ designar, representando 15.2% do serviço docente total do ciclo de estudos. Referem-se a 4 docentes com processos de contratação em curso para o DEI (1 Catedrático, 1 Associado e 2 Auxiliares), responsáveis por 6,3% do serviço docente, um número estimado de 5 Professores Auxiliares convidados (pos-docs), responsáveis por 3,5% do serviço docente, e um número estimado de 9 Assistentes convidados (estudantes de doutoramento), responsáveis por 5,3% do serviço docente.*

##### 5.6. Observations:

a) In Table 5.3, it are indicated the teachers currently integrated in the teaching or research career, as well as some invited teachers with a contract, that are expected to be involved in the teaching of the study cycle, in a scenario where the volume of students is similar to that of the 1st cycle (1st to 3rd year) of the current integrated study cycles (MIEIC and MIERSI), with 2 theoretical turns and 9 to 13 practical turns per UC. The time regime refers to the link with UP, and not to the proportion of dedication to this particular study cycle; in fact, in most cases, teachers collaborate in multiple study cycles.

b) The teachers allocated to teach UCs classified in the fundamental area of the study cycle (EIC) are specialized in EIC. The teachers allocated to teach UCs in the areas of Mathematics, Physics and Transferrable Skills are not specialized in EIC (they are specialized in the areas of those UCs).

c) In some curricular units, there are hours of teaching service allocated to teachers to hire / appoint, representing 15.2% of the total teaching service in the study cycle. They refer to 4 professors with ongoing hiring processes for the DEI (1 Full Professor, 1 Associate Professor, and 2 Assistants Professors), responsible for 6.3% of the teaching service, an estimated number of 5 Invited Assistant Professors (postdocs), responsible for 3.5% of the teaching service, and an estimated number of 9 Invited Assistants (PhD students), responsible for 5.3% of the teaching service.

## 6. Pessoal Não Docente

### 6.1. Número e regime de tempo do pessoal não-docente afeto à lecionação do ciclo de estudos.

• *Serviços Académicos (SERAC) - 18 recursos humanos (RH) a tempo inteiro dão apoio transversal a todos os ciclos de estudos (CEs) da FEUP nas seguintes áreas: administração e apoio à gestão; acesso, ingresso e certificação; gestão de estudantes, de acordo com as instruções e as diretivas dos Órgãos de Gestão, constituindo a relação com o estudante o vetor essencial da sua atuação.*

• *Departamento de Engenharia Informática (DEI) – 3 RH parcialmente afetos ao CE que garantem as atividades de administração, gestão e secretariado.*

• *Gabinete de Sistemas de Informação (GSI) – 6 RH que dão apoio de laboratório a tarefas de ensino, investigação e serviços, no âmbito das atividades desenvolvidas no CE.*

• *Departamento de Ciência de Computadores (DCC) - 2 técnicos superiores, responsáveis pela administração e manutenção da rede informática, laboratórios e servidores; 2 assistentes técnicas, responsáveis pelo secretariado; 1 auxiliar técnica, responsável pela manutenção das instalações.*

### 6.1. Number and work regime of the non-academic staff allocated to the study programme.

• *Academic Services (SERAC) - 18 full-time human resources (HR) that provide cross-cutting support to all FEUP study cycles (SCs) in the following areas: administration and management support; access, admission and certification; student management, in accordance with the instructions and guidelines of the management bodies, making the relationship with the student the essential vector of its performance.*

• *Department of Informatics Engineering (DEI) - 3 HR partially assigned to the SC, ensuring administration, management and secretarial activities.*

• *Information Systems Office (GSI) - 6 HR that provide laboratory support for teaching, research and service tasks in the scope of the SC activities.*

• *Department of Computer Science (DCC) - 2 senior technicians, responsible for the administration and maintenance of the computer network, laboratories and servers; 2 technical assistants, responsible for the secretariat; 1 technical assistant, responsible for the maintenance of the facilities.*

### 6.2. Qualificação do pessoal não docente de apoio à lecionação do ciclo de estudos.

*Dos 18 recursos humanos afetos aos Serviços Académicos, 3 possuem mestrado, 11 licenciatura e 4 o ensino secundário. O número de recursos humanos dos Serviços Académicos com formação superior ajusta-se ao aumento de complexidade do serviço e às suas necessidades, tendo-se verificado uma evolução em termos de habilitações, que se reflete indiretamente na qualidade do trabalho realizado.*

*Dos 3 recursos humanos afetos ao Departamento de Engenharia Informática (DEI), dois possuem licenciatura e outro ensino secundário.*

*Dos 6 colaboradores afetos ao Gabinete de Sistemas de Informação, 2 possuem mestrado, 3 licenciatura e 1 o 12º ano. Os 5 técnicos do DCC têm as seguintes habilitações: 2 técnicos superiores (mestres); 2 assistentes técnicas (licenciadas); 1 auxiliar técnica (6º ano).*

### 6.2. Qualification of the non-academic staff supporting the study programme.

*Of the 18 human resources allocated to the Academic Services, 3 have master's degree, 11 "licenciatura" and 4 high school diplomas. The number of human resources of the Academic Services with higher education is adjusted to the increasing complexity of the service and its needs. There has been an evolution in terms of qualifications, which is indirectly reflected in the quality of the work performed.*

*Of the 3 human resources allocated to the Department of Informatics Engineering (DEI), two have "licenciatura" and the other a high school diploma.*

*Of the 6 human resources allocated to the Information Systems Office, 2 have a master's degree, 3 have a "licenciatura" degree and 1 has the 12th year.*

*The 5 DCC technicians have the following qualifications: 2 senior technicians (masters); 2 technical assistants (licensed); 1 technical assistant (6th year).*

### 6.3. Procedimento de avaliação do pessoal não-docente e medidas conducentes à sua permanente atualização e desenvolvimento profissional.

*A avaliação de desempenho do Pessoal Não Docente obedece a metodologia e a critérios pré-definidos. Os Trabalhadores em exercício de funções públicas são avaliados de acordo com o Sistema Integrado de Avaliação de Desempenho na Administração Pública (SIADAP), e os Trabalhadores com contrato em regime de direito privado da UP são avaliados de acordo com o Sistema de Avaliação de Desempenho da Universidade do Porto (SIADUP). Promove-se a avaliação de desempenho como instrumento de reconhecimento do mérito e de melhoria da qualidade. Anualmente é definido o plano de formação do pessoal não docente que resulta de levantamentos de necessidades de formação, no sentido de melhorar as qualificações do pessoal não docente.*

### 6.3. Assessment procedures of the non-academic staff and measures for its permanent updating and personal development

*The performance evaluation of Non-Teaching Staff follows the pre-defined methodology and criteria. Employees in public service are assessed in accordance with the Integrated System of Performance Assessment in Public Administration (SIADAP), and Workers with contract under the private law of UP are assessed in accordance with the Performance Evaluation System of Public Administration of University of Porto (SIADUP). Performance evaluation is promoted as a tool for recognizing merit and improving quality. The training plan for non-teaching staff that results from training needs assessments to improve the qualifications of non-teaching staff is defined annually.*

## 7. Instalações e equipamentos

### 7.1. Instalações físicas afetas e/ou utilizadas pelo ciclo de estudos (espaços letivos, bibliotecas, laboratórios, salas de computadores, etc.):

*As instalações físicas afetas ao ciclo de estudos no edifício da FEUP incluem:*

- *Biblioteca - Gabinetes de estudo individual e salas de leitura (27 salas)*
- *Sala de Aula - Anfiteatro (3 salas de 184 lugares; 13 salas de 99 lugares; 18 salas de 53 a 60 lugares)*
- *Sala de Aula - Computadores (3 salas de 16 a 20 lugares; 14 salas de 24 a 27 lugares)*
- *Sala de Aula - Teórico Práticas (34 salas de 25 a 46 lugares)*
- *Laboratório de Projeto - Acesso Livre (1 sala de 20 lugares)*
- *Salas de Exame (12 salas de 44 a 89 lugares)*
- *Labs de I&D: Computação Gráfica, Interação e Jogos; Computação Sonora e Musical; Engenharia de Software; Inteligência Artificial – Robótica e Simulação; Inteligência Artificial – Sistemas Multi-Agente; Sistemas de Computação; Sistemas de Informação (7 salas de 20 a 24 lugares)*
- *Gabinetes para atendimento de estudantes pela equipa docente da FEUP e FCUP*

### 7.1. Facilities used by the study programme (lecturing spaces, libraries, laboratories, computer rooms, ...):

*The physical facilities allocated to the cycle of studies at the FEUP campus include:*

- *Library - Individual study offices and reading rooms (27 rooms)*
- *Classroom - Amphitheater (3 rooms with 184 seats; 13 rooms with 99 seats; 18 rooms with 53 to 60 seats)*
- *Classroom - with Computers (3 rooms with 16 to 20 seats; 14 rooms with 24 to 27 seats)*
- *Classroom - for both lectures and practical classes (34 rooms with 25 to 46 seats)*
- *Project Lab - Open Access (1 room with 20 seats)*
- *Examination rooms (12 rooms with 44-89 seats)*
- *R&D Labs: Artificial Intelligence– Multi-Agent Systems; Artificial Intelligence– Robotics and Simulation; Computing Systems; Information Systems; Graphics, Interaction and Games; Software Engineering; Sound and Music Computing (7 rooms with 20 to 24 seats)*
- *Various offices available to receive students, that can be used by FEUP and FCUP teaching staff*

### 7.2. Principais equipamentos e materiais afetos e/ou utilizados pelo ciclo de estudos (equipamentos didáticos e científicos, materiais e TIC):

*Os principais equipamentos e materiais afetos ao ciclo de estudos incluem:*

- *Salas de Aula - Computadores Pessoais, Projetores de Vídeo, Equipamentos de Rede.*
- *Laboratório de Projeto, Acesso Livre – Computadores.*
- *Biblioteca - Editoras de Revistas Científicas; Bases de Dados Bibliográficas; Enciclopédias e Dicionários; Editoras de e-Books; Bibliotecas Repositórios Digitais; Livros de Texto; Revistas em papel; Títulos com acesso on-line.*
- *Centro de Informática - Pontos de Acesso à Rede (cabo); Pontos de Acesso à Rede (wireless); Impressoras em rede; Servidores; Routers; Switches; Cluster com interligação em GRID (64 processadores).*
- *Labs. de I & D – Computadores (PCs e servidores); Robôs; Dispositivos IoT (sensores & atuadores); “Wearables”.*

### 7.2. Main equipment or materials used by the study programme (didactic and scientific equipment, materials, and ICTs):

*The main equipment and materials allocated the study course include:*

- *Classrooms - Personal Computers, Video Projectors, Network Equipment.*
- *Project Lab - Computers*
- *Library - access to Scientific Journals, Bibliographic Databases, Encyclopedias and Dictionaries, e-Books, Digital Repositories and Libraries Text Books, hardcopy Magazines, Online articles.*
- *Informatics Centre - cable and wireless network access points; network printers; servers, routers; switches; grid computing (64 processors)*
- *R&D Labs – Computers (PCs and servers); Robots; IoT equipment (sensors & actuators); Wearables*

## 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
INESC TEC - Tecnologia e Ciência / INESC TEC Institute for Systems and Computer Engineering, Technology and Science	Muito Bom	Inesc Tec - Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência (INESC TEC)	60	Resultados da Avaliação de Unidades I&D 2017/2018
LIACC- Laboratório de Inteligência Artificial e Ciência de Computadores / Artificial Intelligence and Computer Science Laboratory	Excelente	U.Porto	11	Resultados da Avaliação de Unidades I&D 2017/2018
Centro de Matemática da Universidade do Porto (CMUP) / The Centre of Mathematics of the University of Porto	Excelente	U. Porto	8	Resultados da Avaliação de Unidades I&D 2017/2018
Laboratório Associado de Energia, Transportes e Aeronáutica (LAETA) / LAETA - ASSOCIATED LABORATORY FOR ENERGY, TRANSPORTS AND AERONAUTICS	Excelente	INEGI - Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI/UP)	7	Resultados da Avaliação de Unidades I&D 2017/2018
Instituto de Telecomunicações (IT) / Telecommunications Institute	Muito Bom	Instituto de Telecomunicações / Telecommunications Institute	6	Resultados da Avaliação de Unidades I&D 2017/2018
Centro de Física das Universidades do Minho e do Porto (CF-UM-UP) / Centre of Physics of the University of Minho and Porto	Muito Bom	U. Porto, U. Minho	2	Resultados da Avaliação de Unidades I&D 2017/2018
Centro de Física da Universidade de Coimbra (CFisUC) / Center for Physics of the University of Coimbra	Muito Bom	U. Coimbra	2	Resultados da Avaliação de Unidades I&D 2017/2018
Centro de Investigação em Sistemas Computacionais Embebidos e de Tempo-Real (CISTER) / Research Centre in Real-Time and Embedded Computing Systems	Excelente	Instituto Superior de Engenharia do Porto (ISEP/IPP)	1	Resultados da Avaliação de Unidades I&D 2017/2018
Centro de Sistemas e Tecnologias (SYSTEC) / Research Center for Systems and Technologies	Muito Bom	Faculdade de Engenharia da Universidade do Porto (FE/UP)	1	Resultados da Avaliação de Unidades I&D 2017/2018
Centro de Estudos de Fenómenos de Transporte (CEFT) / Transport Phenomena Research Center	Excelente	Faculdade de Engenharia da Universidade do Porto (FE/UP)	1	Resultados da Avaliação de Unidades I&D 2017/2018
Centro de Investigação da Terra e do Espaço da Universidade de Coimbra (CITEUC) / Centre for Earth and Space Research of the University of Coimbra	Bom	U. Coimbra	1	Resultados da Avaliação de Unidades I&D 2017/2018
Centro de Investigação em Tecnologias e Serviços de Saúde (CINTESIS) / Center for Health Technology and Services Research	Muito Bom	Faculdade de Medicina da Universidade do Porto (FM/UP)	1	Resultados da Avaliação de Unidades I&D 2017/2018

### Pergunta 8.2. a 8.4.

#### 8.2. Mapa-resumo de publicações científicas do corpo docente do ciclo de estudos, em revistas de circulação internacional com revisão por pares, livros ou capítulos de livro, relevantes para o ciclo de estudos, nos últimos 5 anos.

<http://www.a3es.pt/si/iportal.php/cv/scientific-publication/formId/52badfdd-f450-93e4-62f0-5ed613d0b264>

#### 8.3. Mapa-resumo de atividades de desenvolvimento de natureza profissional de alto nível (atividades de desenvolvimento tecnológico, prestação de serviços ou formação avançada) ou estudos artísticos, relevantes para o ciclo de estudos:

<http://www.a3es.pt/si/iportal.php/cv/high-level-activities/formId/52badfdd-f450-93e4-62f0-5ed613d0b264>

#### 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 corpo docente da LiEIC participa e/ou lidera vários projetos de investigação e desenvolvimento e tem diversas relações internacionais.*

*Alguns projetos em que membros do corpo docente da LiEIC participam:*

- ANTAREX - AutoTuning and Adaptivity appRoach for Energy efficient eXascale HPC systems (H2020 – 671623)
- CHIC (POCI-01-0247-FEDER-024498)
- BEACONING - Breaking Educational Barriers with Contextualised, Pervasive and Gameful Learning (H2020 - 687676)
- FEEdBACK - Fostering Energy Efficiency and BehAvioural Change through ICT (H2020 - 768935)
- RECAP - Research on European Children and Adults born Preterm (H2020 - 733280)
- AutoDriving (POCI-01-0145-FEDER-028526)
- PRODUTECH SIF - Soluções para a Indústria de Futuro (POCI-01-0247-FEDER-024541)
- INDTECH 4.0 (POCI-01-0247-FEDER-026653)
- C-ROADS PORTUGAL (72070, 2016-PT-TM-0259-S, CE)
- Simusafe – Simulator of Behavioural Aspects for Safer Transport (H2020 – 723386)
- URready4OS - Expanded Underwater Robotics ready for Oil Spill (H2020 - ECHO/SUB/2016/740129/PREP/21)
- IA.SAE – Inteligência Artificial na Segurança Alimentar e Económica (FCT)
- DARGMINTS (FCT - POCI-01-0145-FEDER-031460)
- EPISA - Inferência de entidades e propriedades para arquivos semânticos (FCT - DSAIPA/DS/0023/2018)
- IntellWheels2.0 – (POCI-01-0247-FEDER-39898)
- opti-MOVES (POCI-01-0145-FEDER-032053)
- Sono ao Volante 2.0 – (NORTE-01-0247-FEDER-039720)
- PROject ManagEment intelligent aSSistAnt (NORTE-01-0247-FEDER-039887)
- RECLAIM - RE-manufaCturing and Refurbishment LArge Industrial equipment (H2020 – 869884)
- HADES: HARdware-backed trusted and scalable DEcentralized Systems (PTDC/CCI-INF/31698/2017)
- Estetoscópio Digital para Uso Clínico (PTDC/CCI-COM/29200/2017)
- FGPE - (2018-1-PL01-KA203-050803)/Erasmus+
- M-BW (PT/FB/BL-2016-312)/BIAL
- DIDIMO (H2020-SMEInst-2018-2020-2, UP spin-off)
- Text2Story (PTDC/CCI-COM/31857/2017)
- Human-AI (H2020 820437)
- MobiWise (POCI-01-0145-FEDER-016426)
- Safe Cities - Inovação para Construir Cidades Seguras (POCI-01-0247-FEDER-041435)
- LastMile (POCI-01-0145-FEDER-028611)
- AVC em translação (PTDC/MEC-NEU/31674/2017)
- NaSA (PTDC/MEC-ONC/32018/2017)
- Solving the 3D chromatin structure of CDH (PTDC/BTM-TEC/30164/2017)
- Tregs in cancer immune response (PTDC/MED-PAT/32462/2017)
- GenomePT (POCI-01-0145-FEDER-022184)
- TRINEV (H2020-MSCA-ITN-2016/H2020-MSCA-ITN-2016, Grant n° 722148)

*Algumas parcerias:*

- Universidades de Lisboa, Coimbra, Aveiro, Minho, UNL, IUL, UTAD, UAçores, P.Porto, IPB, IPCA, IPVC, São Paulo, Federal de Pernambuco, UFSC, Carnegie Mellon, MIT, Graz, Sheffield, Luxemburg, Delft, UTexas, South Australia, Naples Federico II, Milano, TU Darmsta
- Palo Alto Research Center, CERN, Microsoft Research, Institute of Science and Technology (ESEO), Google, Google Deepmind, TAP, Facebook, HondaRI, Schneider, APTIV, SONAE IM, Delloite, Volkswagen, IBM, Altran, Fraunhofer Portugal, Bosch Car, Continental Mabor, Critical Manufacturing, EFACEC, Farfetch, Feedzai, Natixis, NOS

#### 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.

*LiEIC faculty members are participants and/or leaders in several research and development projects and have several international relationships with other institutions.*

*Some projects in which members of LiEIC faculty staff participate: :*

- ANTAREX-AutoTuning and Adaptivity appRoach for Energy efficient eXascale HPC systems (H2020 – 671623)
- CHIC(POCI-01-0247-FEDER-024498)
- BEACONING - Breaking Educational Barriers with Contextualised, Pervasive and Gameful Learning (H2020 - 687676)
- FEEdBACK - Fostering Energy Efficiency and BehAvioural Change through ICT (H2020 - 768935)
- RECAP - Research on European Children and Adults born Preterm (H2020 - 733280)
- AutoDriving (POCI-01-0145-FEDER-028526)
- PRODUTECH SIF - Soluções para a Indústria de Futuro (POCI-01-0247-FEDER-024541)
- INDTECH 4.0 (POCI-01-0247-FEDER-026653)
- C-ROADS PORTUGAL (72070, 2016-PT-TM-0259-S, CE)
- Simusafe – Simulator of Behavioural Aspects for Safer Transport (H2020 – 723386)
- URready4OS - Expanded Underwater Robotics ready for Oil Spill (H2020 - ECHO/SUB/2016/740129/PREP/21)
- IA.SAE – Artificial Intelligence for Food and Economic Safety (FCT)
- DARGMINTS (FCT - POCI-01-0145-FEDER-031460)
- EPISA - Inferência de entidades e propriedades para arquivos semânticos (FCT - DSAIPA/DS/0023/2018)
- IntellWheels2.0 – (POCI-01-0247-FEDER-39898)
- opti-MOVES (POCI-01-0145-FEDER-032053)
- Sono ao Volante 2.0 – (NORTE-01-0247-FEDER-039720)
- PROject ManagEment intelligent aSSistAnt (NORTE-01-0247-FEDER-039887)
- RECLAIM - RE-manufaCturing and Refurbishment LArge Industrial equipment (H2020 – 869884)
- HADES: HARdware-backed trusted and scalable DEcentralized Systems (PTDC/CCI-INF/31698/2017)
- Estetoscópio Digital para Uso Clínico (PTDC/CCI-COM/29200/2017)

- FGPE - (2018-1-PL01-KA203-050803)/Erasmus+
- M-BW (PT/FB/BL-2016-312)/BIAL
- DIDIMO (H2020-SMEInst-2018-2020-2, UP spin-off)
- Text2Story (PTDC/CCI-COM/31857/2017)
- Human-AI (H2020 820437)
- MobiWise (POCI-01-0145-FEDER-016426)
- Safe Cities - Inovação para Construir Cidades Seguras (POCI-01-0247-FEDER-041435)
- LastMile (POCI-01-0145-FEDER-028611)
- AVC em translação (PTDC/MEC-NEU/31674/2017)
- NaSA (PTDC/MEC-ONC/32018/2017)
- Solving the 3D chromatin structure of CDH (PTDC/BTM-TEC/30164/2017)
- Tregs in cancer immune response (PTDC/MED-PAT/32462/2017)
- GenomePT (POCI-01-0145-FEDER-022184)
- TRAINEV (H2020-MSCA-ITN-2016/H2020-MSCA-ITN-2016, Grant nº 722148)

Some partnerships:

Universities of Lisbon, Coimbra, Aveiro, Minho, New Lisbon, IUL, UTAD, UAzores, P.Porto, IPB, IPCA, IPVC, São Paulo, Federal of Pernambuco, UFSC, Carnegie Mellon University, MIT, Graz, Sheffield, Luxemburg, Delft, UTexas, South Australia, Naples Federico II, Milano, TU Darmstadt

•Palo Alto Research Center, CERN, Microsoft Research, Institute of Science and Technology (ESEO), Google, Google Deepmind, TAP, Facebook, HondaRI, Schneider, APTIV, SONAE IM, Delloite, Volkswagen, IBM, Altran, Fraunhofer Portugal, Bosch Car, Continental Mabor, Critical Manufacturing, EFACEC, Farfetch, Feedzai, Natixis, NOS

## 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 dados da DGES, a percentagem de recém-diplomados (2014-2017) que em 2018 se encontravam registados como desempregados no IEFP era de 0,9%/1,1% para diplomados do MIEIC/MIERSI e 1,5% para diplomados na mesma área de formação em todo o ensino superior público.*

*De acordo com o estudo “Digital organisational frameworks and IT professionalism” da Comissão Europeia de março de 2019, num cenário de crescimento moderado, o nº de profissionais de TIC na EU-28 em 2020 será de cerca de 6 milhões (75% dos quais em funções de gestão, análise, arquitetura, desenvolvimento e administração), para uma procura de 6.4 milhões, resultando num deficit de 400 mil profissionais.*

*O inquérito de empregabilidade realizado pela FEUP em 2019 a diplomados do MIEIC no ano letivo 2017/2018 indica que 96% obtiveram emprego até 10 meses após a conclusão do ciclo de estudo (59% antes da graduação) em funções total ou parcialmente relacionadas com a área de formação, tendo outros 3% iniciado nova formação.*

### 9.1. Evaluation of the employability of graduates by similar study programmes, based on official data:

*According to DGES data, the percentage of recent graduates (2014-2017) who in 2018 were registered as unemployed at IEFP was 0.9%/1.1% for MIEIC/MIERSI graduates and 1.5% for graduates in the same area of education throughout public higher education.*

*According to the “Digital organizational frameworks and IT professionalism” study of the European Commission, March 2019, in a scenario moderate growth, the number of ICT professionals in the EU-28 in 2020 will be around 6 million (75% of which in management, analysis, architecture, development and administration jobs), to a demand of 6.4 million, resulting in a deficit of 400 thousand professionals.*

*The 2019 FEUP employability survey of MIEIC graduates in 2017/2018 indicates that 96% were employed within 10 months of graduation (59% even before graduation) in functions totally or partially related with their area of education, and 3 others % started new training.*

### 9.2. Avaliação da capacidade de atrair estudantes baseada nos dados de acesso (DGES):

*Na 1ª fase do concurso nacional de acesso de 2019, o MIEIC teve 867 candidatos para 135 vagas, dos quais 343 em 1ª opção e 196 em 1ª opção com classificação  $\geq 170$ , correspondente a um índice de excelência de 1,45. A classificação do último colocado pelo contingente geral foi de 178,3, a mais elevada do país na área de Engenharia Informática (a 6,3 pontos do ciclo de estudo seguinte) e a 14ª mais elevada no conjunto de todos os ciclo de estudo. A percentagem de colocados em 1ª opção foi de 83% e a de colocados do sexo feminino foi de 24%.*

*Adicionalmente, há uma procura significativa por via de concursos locais, tendo-se verificado em 2019 mais 45 ingressos para o 1º ciclo do MIEIC (concurso especial de estudantes internacionais, titulares de cursos de especialização tecnológica, mudança de par instituição/curso, titulares de outros cursos superiores, regimes especiais), apesar de alguns requisitos de candidatura exigentes.*

*Espera-se que estes níveis de procura se mantenham na LiEIC.*

### 9.2. Evaluation of the capability to attract students based on access data (DGES):

*In the first phase of the 2019 national access application, MIEIC had 867 candidates for 135 vacancies, of which 343 in 1st option, and 196 in 1st option with a grade  $\geq 170$ , corresponding to an excellence index of 1.45. The grade of the last placed by the general contingent was 178.3, the highest in the country in the area of Informatics Engineering (6.3 points above the next course) and the 14th highest among all courses. The percentage of those placed in 1st option was 83% and the percentage of female placed was 24%.*

*In addition, there is a significant demand through local applications, with 45 additional admissions in 2019 for the 1st*

*cycle of MIEIC (special admission for international students, holders of technological specialization courses, change of institution/course, holders of degrees in other areas, special regimes), despite some demanding application requirements.*

*These demand levels are expected to be maintained for LiEIC.*

### **9.3. Lista de eventuais parcerias com outras instituições da região que lecionam ciclos de estudos similares:**

*O Departamento de Engenharia Informática (DEI) da FEUP e o Departamento de Ciência de Computadores (DCC) da FCUP, que lecionam a generalidade das UCs da LiEIC nas áreas científicas da Engenharia Informática, têm ligações a nível de ensino (intercâmbio de docentes, ciclos de estudos em parceria, palestras convidadas, cursos curtos, concursos de programação, etc.) e investigação (em projetos e unidades de I&D) com outras instituições de ensino superior da região (nomeadamente com as Universidades do Minho, UTAD e Aveiro e o Instituto Politécnico do Porto) e com outras escolas da UP (nomeadamente Bela Artes, Letras e Economia), as quais terão impacto na LiEIC. A nível do espaço europeu, o MIEIC tem presentemente (2019/20) acordos bilaterais ERASMUS+ para mobilidade de estudantes com 36 universidades prestigiadas em 16 países europeus, esperando-se que grande parte desses acordos sejam renovados para permitir mobilidade de estudantes da LiEIC tipicamente no 1º semestre do 3º ano.*

### **9.3. List of eventual partnerships with other institutions in the region teaching similar study programmes:**

*The Department of Informatics Engineering (DEI) of FEUP and the Department of Computer Science (DCC) of FCUP, which teach the majority of LiEIC UCs in the scientific areas of Informatics Engineering, has several links in terms of education (exchange of teachers, offer of study cycles in partnership, invited lectures, short courses, programming competitions, etc.) and research (in research projects and research units) with other higher education institutions in the region (namely the Universities of Minho, UTAD and Aveiro and the Polytechnic Institute of Porto) and with other schools of the UP (notably with the Faculties of Fine Arts, Letters and Economics), which will impact LiEIC. At the European level, MIEIC currently has (2019/20) ERASMUS+ bilateral student mobility agreements with 36 prestigious universities in 16 European countries, and it is expected that most of these agreements will be renewed to allow LiEIC student mobility typically in the 1st semester of the 3rd year.*

## **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:**

*No top 100 do “QS World University Rankings 2018”, área de “Computer Science & Information Systems”, encontram-se instituições com CEs de duração igual e estrutura similar à LiEIC (em graus variados), como:*

- *TU Munich – BSc in Informatics – 6 semestres; UCs de 5 a 8 ECTS similares às da LiEIC; UCs optativas de soft skills similares às Competências Transversais na LiEIC; projeto e tese finais no 3º ano (com características similares às modalidades previstas no Projeto Integrador da LiEIC).*
  - *ETH Zurich – B. in CS – 6 semestres; UCs de 5 a 7 ECTS em 3 áreas (systems & software eng., information & data processing, theoretical computer science) que mapeiam para áreas científicas da LiEIC; tese final.*
  - *TU Delft – B. of CS and Eng. - 12 trimestres; UCs de 5 ECTS em 6 áreas (Mathematics, Systems, Models, Software, Data, Media & AI) que mapeiam para áreas da LiEIC; projeto final no último trimestre.*
- Outros exemplos: Cambridge; Oxford; EPFL; Poli. Milano; KU Leuven; KTH; TU Wien.*

### **10.1. Examples of study programmes with similar duration and structure offered by reference institutions in the European Higher Education Area:**

*In the top 100 of the QS World University Rankings 2018, area of Computer Science & Information Systems, one can find several institutions with CEs of equal duration and similar structure to LiEIC (to varying degrees), such as:*

- *TU Munich - BSc in Informatics - 6 semesters; UCs of 5 to 8 ECTS similar to those of LiEIC; elective UCs of soft skills similar to LiEIC's Transversal Skills; final project and thesis in the 3rd year (with characteristics similar to the modalities foreseen in the LiEIC's Capstone Project).*
  - *ETH Zurich - B. in CS - 6 semesters; UCs of 5 to 7 ECTS in 3 areas (systems & software eng., Information & data processing, theoretical computer science) mapping to LiEIC scientific areas; final thesis.*
  - *TU Delft - B. of CS and Eng. - 12 quarters; 5 ECTS UCs in 6 areas (Mathematics, Systems, Models, Software, Data, Media & AI) mapping to LiEIC areas; final project in the last quarter.*
- Other examples: Cambridge; Oxford; EPFL; Poly. Milano; KU Leuven; KTH; Wien.*

### **10.2. Comparação com objetivos de aprendizagem de ciclos de estudos análogos existentes em instituições de referência do Espaço Europeu de Ensino Superior:**

*A maioria dos programas de bacharelato em Engenharia Informática e Computação na Europa leva em consideração os “Curriculum Guidelines for Undergraduate Degree Programs in Computer Science” da ACM e IEEE.*

*Consequentemente, os resultados de aprendizagem e unidades de conhecimento nucleares propostos nessas diretrizes estão geralmente presentes, com variações geralmente associadas a unidades de conhecimento e resultados de aprendizagem considerados opcionais nessas diretrizes. Outras diferenças estão relacionadas com o contexto em que os programas são oferecidos (nacional, regional e local). Por exemplo, o BSc in Informatics da TU Munich propõe o desenvolvimento de conhecimentos e habilidades muito similares à LiEIC. Uma diferença que se encontra em relação a muitos programas na Europa, é uma maior ênfase nas ciências básicas da engenharia, como Matemática e Física, preparando os diplomados para trabalhar em equipas multidisciplinares com colegas de outras áreas de ciência e engenharia.*

## 10.2. Comparison with the intended learning outcomes of similar study programmes offered by reference institutions in the European Higher Education Area:

*Most undergraduate degree programs in Informatics Engineering and Computer Science in Europa take into consideration the "Curriculum Guidelines for Undergraduate Degree Programs in Computer Science" from ACM and IEEE. Consequently, the core knowledge units and learning outcomes proposed in those guidelines are generally found in those programmes, with variations usually associated with knowledge units and learning outcomes considered elective in those guidelines. Other differences are related with the context in which the programmes are offered (national, regional and local). For example, the BSc in Informatics at TU Munich offers the development of expertise and skills very similar to LiEIC. A difference that can be found in comparison to many programs in Europe, is a greater emphasis on basic engineering sciences, such as Mathematics and Physics, preparing graduates to work on multidisciplinary environments with colleagues from other areas of science and engineering.*

## 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 - Protocolos de Cooperação

11.1.1. Entidade onde os estudantes completam a sua formação:

<sem resposta>

11.1.2. Protocolo (PDF, máx. 150kB):

<sem resposta>

11.2. Plano de distribuição dos estudantes

11.2. Plano de distribuição dos estudantes pelos locais de estágio e/ou formação em serviço demonstrando a adequação dos recursos disponíveis.(PDF, máx. 100kB).

<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:

<sem resposta>

11.3. Institution's own resources to effectively follow its students during the in-service training periods:

<no answer>

### 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:

- Elevada procura por estudantes nacionais e internacionais, com 196 candidatos em 1ª opção com nota  $\geq 170$  para 135 vagas pelo concurso nacional de acesso e 45 ingressos por concursos locais em 2019;
- Reforço significativo de *numerus clausus* (135+70), resultante da colaboração entre FEUP e FCUP;
- Elevada empregabilidade dos graduados em Engenharia Informática;
- Elevada reputação da FEUP e FCUP a nível nacional e internacional;
- Formação de banda larga adequada ao mercado e à prossecução de estudos mais avançados;
- Formação com grande componente prática em competências técnicas e soft skills;
- Corpo docente com grande experiência de ensino e I&D&T, cobrindo as áreas nucleares da informática;
- Secretariado com elevada experiência;
- Elevado nº de acordos de mobilidade, incluindo acordos com 36 universidades europeias prestigiadas;
- Localização num campus (Porto Innovation District) que reúne instituições de ensino superior, institutos de I&D, incubadoras e empresas tecnológicas.

### 12.1. Strengths:

- High demand by national and international students, with 196 candidates in 1st option with grade  $\geq 170$  for 135 vacancies in the national access applications and 45 admissions in local applications in 2019;
- Significant reinforcement of *numerus clausus* (135 + 70), resulting from the collaboration between FEUP and FCUP;
- High employability of the graduates in Informatics Engineering;
- High reputation of FEUP and FCUP at national and international level;
- Wideband education suited to market needs and further studies;
- Education with a large practical component in technical and soft skills;
- Faculty with extensive teaching and R&D&T experience covering the core areas of informatics;
- Secretarial with high experience;
- High number of mobility agreements, including agreements with 36 prestigious European universities;
- Campus location (Porto Innovation District) which brings together higher education institutions, R&D institutes, incubators and technology companies.

### 12.2. Pontos fracos:

- Limitações de espaços disponíveis para ensino, avaliação, realização de trabalhos e investigação;
- Corpo docente dos departamentos de Engenharia Informática (FEUP) e Ciência de Computadores (FCUP) aquém das necessidades e com percentagem reduzida de professores com *tenure*, complementado com recurso a docentes convidados;
- Falta de um edifício próprio para o Departamento de Engenharia Informática;
- O desempenho pedagógico dos docentes é levado em conta de forma insuficiente na sua avaliação global;
- Dificuldade em tornar as aulas teóricas em espaços de aprendizagem efetiva;
- Taxa de respostas aos inquéritos pedagógicos relativamente baixa (cerca de 34%);
- Elevadas taxas de insucesso nos estudantes que ingressam por concursos locais ou contingentes especiais;
- Percentagem reduzida de estudantes do sexo feminino (24% dos ingressos em 2019).

### 12.2. Weaknesses:

- Limitations of spaces available for teaching, assessment, students' work and research;
- Faculty members of the Informatics Engineering (FEUP) and Computer Science (FCUP) departments fall short of needs and have a reduced percentage of *tenure* teachers, complemented by the use of invited teachers;
- Lack of a specific building for the Department of Informatics Engineering;
- Teachers' pedagogical performance is insufficiently taken into account in their overall assessment;
- Difficulty in making lectures effective learning spaces;
- Relatively low response rate to pedagogical surveys (around 34%);
- High failure rates in students entering by local applications or special contingents;
- Reduced percentage of female students (24% of admissions in 2019).

### 12.3. Oportunidades:

- Crescente importância da informática para a sociedade;
- Enorme deficit de profissionais na área em relação às necessidades do mercado;
- Contexto favorável à atração de estudantes internacionais, nomeadamente dos países de língua oficial portuguesa, e para intercâmbio internacional de estudantes;
- Possibilidade de tirar maior partido das tecnologias de informação e metodologias de ensino-aprendizagem centradas no estudante para agilizar/otimizar processos pedagógicos e administrativos;
- Projeto integrador no final do 1º ciclo proporciona uma oportunidade de contacto com a realidade da indústria e de

- estabelecimento de parcerias com empresas e outras instituições;*
- Elevado número de alumni em posições de destaque em empresas nacionais e internacionais.*

### 12.3. Opportunities:

- Growing importance of informatics for society;*
- Huge deficit of professionals in the area in relation to market needs;*
- Favorable context for attracting international students, namely from Portuguese-speaking countries, and for international student exchange;*
- Possibility to take full advantage of information technologies and student-centered education-learning methodologies to streamline/optimize pedagogical and administrative processes;*
- Capstone project at the end of the 1st cycle provides an opportunity for contact with the reality of the industry and for establishing partnerships with companies and other institutions;*
- High number of alumni in prominent positions in national and international companies.*

### 12.4. Constrangimentos:

- Evolução demográfica com redução de estudantes a ingressar no ensino superior;*
- Classificações elevadas de entrada podem afastar estudantes com vocação para a Engenharia Informática;*
- Redução do financiamento para o Ensino Superior Público poderá limitar crescimento e capacidade de dar resposta às necessidades do mercado;*
- Condições salariais proporcionadas pela indústria comparativamente à academia limitam a capacidade de contratação de docentes;*
- Oferta escassa e a custos elevados de alojamentos para estudantes;*
- Distância física e ausência de meios de transporte rápidos entre os campus da FEUP e da FCUP dificultam a mobilidade de docentes e de estudantes;*
- Perspetiva de crescente concorrência no espaço Europeu na área do ensino universitário.*

### 12.4. Threats:

- Demographic evolution with reduction of students entering higher education;*
- High entrance grades can drive away students with a vocation for Informatics Engineering;*
- Reduced funding for Public Higher Education may limit growth and ability to meet market needs;*
- Salary conditions provided by industry compared to academia limit the ability to hire teachers;*
- Scarce and expensive provision of student accommodation;*
- Physical distance and absence of fast means of transport between the FEUP and FCUP campuses hinder the mobility of teachers and students;*
- Perspective of increasing competition in the European area in the field of university education.*

### 12.5. Conclusões:

*A crescente importância da informática para a sociedade, a capacidade instalada e experiência acumulada de mais de 25 anos no ensino de Engenharia Informática na FEUP, a reputação nacional e internacional da FEUP e da FCUP, a conjugação de valências da FEUP e da FCUP, e a elevada procura por estudantes em 1ª opção, permitem-nos ter confiança de que, através da LiEIC, a FEUP e a FCUP continuarão a prestar um contributo decisivo à sociedade na área de Engenharia Informática.*

*Prevê-se que as atuais limitações de espaços sejam ultrapassadas com a construção dos novos espaços já em estudo no campus da FEUP.*

*Espera-se que os planos de contratação de novos docentes pela FEUP e FCUP permitam reforçar o corpo docente dos departamentos de Engenharia Informática (FEUP) e Ciência de Computadores (FCUP) e dessa forma responder adequadamente às necessidades de crescimento da área.*

### 12.5. Conclusions:

*The growing importance of informatics for society, the installed capacity and accumulated experience of more than 25 years in the teaching of Informatics Engineering at FEUP, the congregation of competencies and resources of FEUP and FCUP, the national and international reputation of FEUP and FCUP, and the high demand for students in first choice, allow us to have confidence that through LiEIC FEUP and FCUP will continue to make a decisive contribution to society in the area of Informatics Engineering.*

*It is expected that the current space limitations will be overcome with the construction of new spaces already under study at the FEUP campus.*

*The plans for hiring new teachers by FEUP and FCUP are expected to reinforce the faculty members of the Informatics Engineering (FEUP) and Computer Science (FCUP) departments and thus respond appropriately to the growth needs of the area.*