

NCE/21/2100367 — Apresentação do pedido - Novo ciclo de estudos

1. Caracterização geral do ciclo de estudos

1.1. Instituição de Ensino Superior:

Universidade De Coimbra

1.1.a. Outras Instituições de Ensino Superior (em associação) (Decreto-Lei n.º 74/2006, de 24 de março, na redacção conferida pelo Decreto-Lei n.º 65/2018, de 16 de agosto, alterado pelo Decreto-Lei n.º 27/2021 de 16 de abril):

1.1.b. Outras Instituições de Ensino Superior (estrangeiras, em associação) (Decreto-Lei n.º 74/2006, de 24 de março, na redacção conferida pelo Decreto-Lei n.º 65/2018, de 16 de agosto, alterado pelo Decreto-Lei n.º 27/2021 de 16 de abril):

<sem resposta>

1.1.c. Outras Instituições (em cooperação) (Lei n.º 62/2007, de 10 de setembro ou Decreto-Lei n.º 74/2006, de 24 de março, na redacção conferida pelo Decreto-Lei n.º 65/2018, de 16 de agosto):

<sem resposta>

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

Faculdade De Ciências E Tecnologia (UC)

1.2.a. Identificação da(s) unidade(s) orgânica(s) da(s) entidade(s) parceira(s) (faculdade, escola, instituto, etc.) (proposta em associação). (Decreto-Lei n.º 74/2006, de 24 de março, na redacção conferida pelo Decreto-Lei n.º 65/2018, de 16 de agosto, alterado pelo Decreto-Lei n.º 27/2021 de 16 de abril):

1.2.b. Identificação da(s) unidade(s) orgânica(s) da(s) entidade(s) parceira(s) (faculdade, escola, instituto, etc.) (proposta em associação com IES estrangeiras). (Decreto-Lei n.º 74/2006, de 24 de março, na redacção conferida pelo Decreto-Lei n.º 65/2018, de 16 de agosto, alterado pelo Decreto-Lei n.º 27/2021 de 16 de abril):

<sem resposta>

1.2.c. Identificação da(s) unidade(s) orgânica(s) da(s) entidade(s) parceira(s) (faculdade, escola, instituto, empresas, etc.) (proposta em cooperação). (Lei n.º 62/2007, de 10 de setembro ou Decreto-Lei n.º 74/2006, de 24 de março, na redacção conferida pelo Decreto-Lei n.º 65/2018, de 16 de agosto):

<sem resposta>

1.3. Designação do ciclo de estudos:

Mestrado em Engenharia de Software

1.3. Study programme:

Master in Software Engineering

1.4. Grau:

Mestre

1.5. Área científica predominante do ciclo de estudos:

Engenharia de Software

1.5. Main scientific area of the study programme:

Software 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):

481

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:

520

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:

N/A

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

60

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

dois semestres

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

two semesters

1.9. Número máximo de admissões proposto:

40

1.10. Condições específicas de ingresso (art.º 3 DL-74/2006, na redação dada pelo DL-65/2018).

Os candidatos serão avaliados e seriadados de acordo com as classificações das componentes profissional e académica tal como listadas abaixo, sendo relevante uma experiência profissional de cinco anos.

a) Titulares de licenciatura ou equivalente legal em Eng^a Informática, Eng^a Eletrotécnica ou em outras áreas da Eng. e das Ciências Exatas e Naturais; b) Titulares de um grau académico superior estrangeiro conferido na sequência de um ciclo de estudos organizado de acordo com os princípios do Processo de Bolonha por um Estado aderente a este Processo, nas áreas referidas na alínea a); c) Titulares de um grau académico superior obtido no estrangeiro, nas áreas referidas na alínea a) que seja reconhecido como satisfazendo os objetivos do grau de licenciado numa das áreas referidas na alínea a); d) Em casos devidamente justificados, os detentores de um currículo escolar ou profissional que seja reconhecido como atestando capacidade para a frequência deste ciclo de estudos pelo CC da FCTUC

1.10. Specific entry requirements (article 3, DL no. 74/2006, as written in the DL no. 65/2018).

The candidates shall be ranked according to their professional and academic competences as listed below, being most relevant the evidence of five years of professional experience.

a) Holders of a degree or legal equivalent in Computer Engineering, Electrotechnical Engineering, or in other areas of Engineering and of Exact and Natural Sciences; b) Holders of a foreign higher academic degree conferred following a 1st cycle of studies organized in accordance with the principles of the Bologna Process by a State adhering to this Process, in the areas referred to in a); c) Holders of a higher academic degree obtained abroad, in the areas referred to in point a) that is recognized as satisfying the objectives of the degree of licensee in one of the areas mentioned by the Scientific Council of FCTUC; d) In duly justified cases, the holders of a school, scientific or professional curriculum, which is recognized as attesting ability to attend this cycle of studies by the SC of FCTUC

1.11. Regime de funcionamento.

Pós Laboral

1.11.1. Se outro, especifique:

Pós-laboral, disponibilizado à distância

1.11.1. If other, specify:

After-labour hours, distance learning

1.12. Local onde o ciclo de estudos será ministrado:

Online, com os conteúdos síncronos disponibilizados a partir da Universidade de Coimbra

1.12. Premises where the study programme will be lectured:

Online contents access, with synchronous delivery from the University of Coimbra.

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

[1.13_Regulamento_805_A_2020_24_09_RAUC_creditacoes_compressed.pdf](#)

1.14. Observações:

Este é um mestrado destinado a profissionais de software no activo, disponibilizado remotamente em horário pós-

laboral e usando aprendizagem baseado em projecto (Studio) de acordo com os seguintes princípios: i) A interação e participação são tecnologicamente mediadas e apoiadas por equipas online de suporte académico e tecnológico; ii) O desenho curricular é orientado para a permitir o acesso sem limites de tempo e lugar aos conteúdos, processos e contextos de ensino e aprendizagem; iii) O modelo pedagógico é especialmente concebido para o ensino e a aprendizagem em ambientes virtuais. As competências pedagógicas específicas para o ensino à distância foram e são asseguradas pela unidade de ensino à distância da Universidade de Coimbra.

Concretamente, ao longo desta formação os estudantes desenvolvem um projecto real para um cliente externo sob supervisão dos docentes. Todos os conteúdos das unidades curriculares estão disponibilizadas online sendo apresentadas e aplicadas no âmbito do Estúdio de Projecto de Software. Embora o projecto seja desenvolvido em equipa para espelhar as condições habituais de um projecto complexo de software, nomeadamente em termos de soft-skills, a avaliação é individual.

Toda a actividade realizada pelos alunos em qualquer unidade curricular é armazenada num repositório online (git, gitHub, gitLab ou BitBucket) que regista as contribuições individuais as quais são regularmente monitorizadas pelos mentores. Nas unidades curriculares com exame a avaliação final é realizada de forma síncrona, seguindo-se uma oral individual sobre as respostas escritas entregues pelos alunos. No caso do Estúdio de Projecto de Software, em pontos chave do projecto os alunos apresentam ao colectivo de mentores (e peritos convidados) o trabalho desenvolvido nas múltiplas dimensões de produto, processo e projeto. A avaliação final de Studio segue as normas regulamentares de defesa de tese.

1.14. Observations:

This master program is designed for working software professionals. Therefore it is delivered online in after-labour hours, and uses project-based learning (the Studio project where the curricular units' contents are applied in practice) according to the following principles: 1) the interaction and participation is technology mediated; 2) the curricular units are designed to support access without time and space constraints; 3) the pedagogical model is specially designed to teaching and learning in virtual environments. The distance-learning specific pedagogic competences are provided by the distance-learning unit of the University of Coimbra.

In more concrete terms, the students develop a real software product for an external client under faculty supervision acting as external consultants. All the curricular units' contents are available online and they are presented and discussed in terms of their application to the Studio. While the project is developed as a team endeavor to mimic a realistic complex software project namely in terms of required soft-skills, the assessment is individual.

All the activities performed by the students are logged in a online repository (git, gitHub, gitLab or Bitbucket) which records every individual contribution. These are regularly monitored by the mentors. In the curricular units with final exam, this final assessment is performed synchronously, followed by oral individual Q&A concerning the answers provided by the students in written form. In the Studio project, at key project milestones, the students present to all the mentors (and invited guests from industry) the work performed in the multiple dimensions of product, project and process. The final Studio assessment conforms to the legal requirements for a Master Thesis defence.

2. Formalização do Pedido

Mapa I - Reitor da Universidade de Coimbra

2.1.1. Órgão ouvido:

Reitor da Universidade de Coimbra

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

[2.1.2._38_2022_Mestrado_Engenharia_de_Software_Im0522pt_en_Modelo_de_despacho_reitoria_signed-compress0.pdf](#)

Mapa I - Conselho Científico da FCTUC

2.1.1. Órgão ouvido:

Conselho Científico da FCTUC

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

[2.1.2._CCcientifico.pdf](#)

Mapa I - Conselhos Pedagógico da FCTUC

2.1.1. Órgão ouvido:

Conselhos Pedagógico da FCTUC

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

[2.1.2._CPedagogico.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:

Este ciclo de estudos tem por objetivo oferecer formação avançada em Engenharia de Software para profissionais no activo fornecendo-lhes as competências mais avançadas que lhes permitirão trabalhar, inovar e gerir projetos de grande dimensão, complexidade e restrições impostas pelo mercado.

Este curso poderá também influenciar de forma profunda e positiva a maneira como a Universidade de Coimbra constrói a sua oferta de mestrados profissionais, pensados para formar líderes técnicos altamente qualificados para a indústria.

3.1. The study programme's generic objectives:

This cycle of studies aims to provide advanced training in Software Engineering for working professionals by providing them the most advanced competences to manage, work and innovate in projects of large size, complexity, and constraints imposed by the market.

This course can also influence in a profound and positive way how the University of Coimbra builds its offer in professional master's degrees, designed to train highly qualified industry leaders.

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

Preparar profissionais para gerir, desenhar e analisar projectos complexos de software realizados por equipas multidisciplinares utilizando uma estratégia de acção reflectida baseada na análise objectiva de factos (e.g. CMMI, SpiCE).

No final deste ciclo de estudos os alunos deverão:

1. Saber gerir o tempo, tarefas, e o trabalho em equipa num projecto de software (o que exige soft skills);
2. Ser capazes de identificar requisitos de cliente e de sistema incluindo as dimensões contextual, legal e regulamentar;
3. Saber desenhar e validar arquitecturas de software capazes de responder às necessidades funcionais e não-funcionais do problema.
4. Ser capazes de adaptar um processo canónico de desenvolvimento de software às circunstâncias concretas do projeto.
5. Saber como verificar e validar (V&V) a adequação da solução técnica ao problema a resolver utilizando as mais recentes metodologias (heurísticas e/ou formais).

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

Prepare professionals to manage, design and analyze complex software projects built by multidisciplinary teams by using a reflexive practice mindset grounded on evidence-based approaches such as CMMI or SpiCE. At the conclusion of this cycle of studies the students must:

- 1) Know how to manage time, activities and technical people in software projects;
- 2) Be able to elicit client and system requirements including the contextual, legal and regulatory dimensions.
- 3) Know how to design and assess software architectures able to address the functional and non-functional requirements of a problem.
- 4) Be able to adapt a canonical software development process to a specific project reality.
- 5) Know how to verify and validate (V&V) the fitness for purpose of the technical solution using the most modern methodologies, using either heuristics and/or formal approaches.

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 Universidade de Coimbra é uma instituição inovadora, virada para o futuro, e atenta aos sinais da sociedade global em que se integra. Deste modo, este Mestrado avançado em Engenharia de Software é uma oportunidade única para ir de encontro à estratégia de inovação da própria instituição. Do ponto de vista de formação, o curso foca-se na formação profissional avançada, a qual é claramente um objectivo da Universidade de Coimbra, que tenta de forma contínua responder às necessidades da indústria nacional e internacional, bem como ao alargamento do paradigma de ensino não restrito ao modelo presencial, acelerado pela pandemia. De modo a satisfazer os objetivos gerais da instituição, este curso rege-se de acordo com o estipulado no "Regulamento de cursos de segundo ciclo na FCTUC". A criação deste curso inovador influenciará decisivamente a forma como a Universidade de Coimbra atualmente vê a sua potencial oferta em termos de mestrados profissionais, isto é, mestrados que, como este, não são pensados apenas para servir de passagem para o doutoramento, mas sobretudo para formar líderes técnicos altamente qualificados para as indústrias do séc. XXI.

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:

The University of Coimbra is an innovative institution that looks into future, while aware of the signals sent by the global society in which it is inserted. Thus, this Advanced Master Degree in Software Engineering is a unique opportunity to meet the innovation strategy of the institution. To match the general objectives of the institution, this program follows the stipulations of the "Rules of the second cycle courses of FCTUC". The implementation of this inovative program shall influence the way the University of Coimbra currently sees its potential to offer other professional masters, i.e. master programs that, like this one, are not thought only to serve as a pathway for doctoral studies, but targetting the training of highly skilled technical leaders for the XXI century industry landscape.

4. Desenvolvimento curricular

4.1. Ramos, variantes, áreas de especialização, especialidades ou outras formas de organização em que o ciclo de estudos se estrutura (a preencher apenas quando aplicável)

4.1. Ramos, variantes, áreas de especialização, especialidades ou outras formas de organização em que o ciclo de estudos se estrutura (a preencher apenas quando aplicável) * / Branches, variants, specialization areas, specialties or other forms of organization (if applicable)*

Ramos, variantes, áreas de especialização, especialidades ou outras formas de organização em que o ciclo de estudos se estrutura * Branches, variants, specialization areas, specialties or other forms of organization

<sem resposta>

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

Mapa II - Engenharia de Software

4.2.1. Ramos, variantes, áreas de especialização, especialidades ou outras formas de organização em que o ciclo de estudos se estrutura (a preencher apenas quando aplicável)*:

Engenharia de Software

4.2.1. Branches, variants, specialization areas, specialties or other forms of organization (if applicable)*

Software Engineering

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

Área Científica / Scientific Area	Sigla / Acronym	ECTS Obrigatórios / Mandatory ECTS	ECTS Mínimos optativos** / Minimum Optional ECTS**	Observações / Observations
Informática	INF	60	0	
(1 Item)		60	0	

4.3 Plano de estudos

Mapa III - Engenharia de Software - 1º e 2º semestres (único ano) / First and second semesters (one only academic year)

4.3.1. Ramos, variantes, áreas de especialização, especialidades ou outras formas de organização em que o ciclo de estudos se estrutura (a preencher apenas quando aplicável)*:

Engenharia de Software

4.3.1. Branches, variants, specialization areas, specialties or other forms of organization (if applicable)*

Software Engineering

4.3.2. Ano/semestre/trimestre curricular:

1º e 2º semestres (único ano) / First and second semesters (one only academic year)

4.3.3 Plano de Estudos / Study plan

Unidade Curricular / Curricular Unit	Área Científica / Scientific Area (1)	Duração / Duration (2)	Horas Trabalho / Working Hours (3)	Horas Contacto / Contact Hours (4)	ECTS Opcional	Observações / Observations
Estúdio de Projeto de Software / Software Project Studio	INF	Anual/Annual	810	OT-60;	30	Aprendizagem baseada em projeto / project- based learning
Gestão do Desenvolvimento de Software / Managing Software Development	INF	Semestral/ Semester	162	T-28; PL-14;	6	-
Métodos: decidir o que construir/ Methods: deciding what to design	INF	Semestral/ Semester	162	T-28; PL-14;	6	-
Modelos de Sistemas de Software / Models of Software Systems	INF	Semestral/ Semester	162	T-28; PL-14;	6	-
Análise de Artefactos de Software / Analysis of Software Artifacts	INF	Semestral/ Semester	162	T-28; PL-14;	6	-
Arquitecturas de Software / Architecture for Software Systems	INF	Semestral/ Semester	162	T-28; PL-14;	6	-

(6 Items)

4.4. Unidades Curriculares

Mapa IV - Estúdio de Projeto de Software

4.4.1.1. Designação da unidade curricular:

Estúdio de Projeto de Software

4.4.1.1. Title of curricular unit:

Software Project Studio

4.4.1.2. Sigla da área científica em que se insere:

INF

4.4.1.3. Duração (anual, semestral ou trimestral):

Anual / Year

4.4.1.4. Horas de trabalho (número total de horas de trabalho):

810

4.4.1.5. Horas de contacto:

OT - 60

4.4.1.6. Créditos ECTS:

30

4.4.1.7. Observações:

O projeto de Estúdio estende-se ao longo todo o ciclo de estudos e representa 50% da carga total de trabalho dos alunos, servindo como contexto onde estes aplicam de forma integrada os conhecimentos adquiridos nas disciplinas do curso, complementadas pelos aspectos humanos inerentes a trabalhar em equipa.

4.4.1.7. Observations:

The Studio project is executed during the whole duration of this cycle of studies and represents 50% of the total workload. It serves as the integration testbed where students apply the knowledge and competences acquired in the other curricular units, complemented by the human aspects intrinsic to a team work.

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

João Gabriel Monteiro Carvalho e Silva; contacto: 31h

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

Qualquer um dos docentes associados a este ciclo de estudos está associado a esta unidade curricular dado que se trata de uma actividade de projecto tutorada. Cada docente deverá assumir a mentoria partilhada de uma equipa, a que correspondem aproximadamente 2h de contacto bi-semanais (os projectos realizam-se normalmente em iterações de duas semanas) complementadas pela análise dos artefactos de projeto produzidos (2h semanais).

António Damasceno; contacto: 2h

Bruno Cabral; contacto: 5h

Henrique Madeira; contacto: 5h

José Pereira; contacto: 2h

Marco Vieira; contacto: 5h

Mário Relá; contacto: 5h

Raul Barbosa; contacto: 5h

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

1. Compreender as actividades envolvidas na gestão de um projeto de software;

2. Saber comunicar informação técnica oralmente e por escrito;

3. Compreender as diferentes etapas da formação da equipa: estilos de liderança, tomada de decisão, abordagens para resolução de conflitos

4. Compreender as fases de desenho de software nomeadamente a definição da arquitectura; saber identificar as diferentes restrições e atributos de qualidade que influenciam esse desenho

5. Saber utilizar as ferramentas e metodologias de uma pipeline de desenvolvimento de software, nomeadamente na integração e entrega contínuos (CI/CD)

6. Compreender os aspectos de garantia de qualidade de um projeto de software

7. Compreender o impacto do factor humano no desenvolvimento de software

O objetivo não é apenas os alunos serem capazes de concluir um projeto técnico para um cliente, mas aprender a fazê-lo com alta qualidade, a reflectir sobre a evolução do projeto e saber justificar adequadamente as decisões tomadas

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

1. Understand all the activities involved in managing software development;

2. Know how to communicate technical content in oral and written form;

3. Understand the different stages of a team, leadership styles, decision taking, conflict resolution techniques;

4. Understand the phases of project design namely architecture definition; be able to identify the different constraints and quality attributes that impact a software design;

5. Know how to use tools and techniques of a software development pipeline, namely continuous integration and deployment (CI/CD);

6. Understand the quality assurance aspects of a software project;

7. Understand the impact of the Human Aspects on Software Engineering.

The course objectives are not limited to successfully build and deploy a technical software artifact, but rather be able to do it with high quality, being able to reflect on the project evolution, and justify sensibly the decisions taken (reflective practice).

4.4.5. Conteúdos programáticos:

Desenvolvimento e gestão de um projeto de desenvolvimento de software para um cliente real: requisitos, arquitetura e desenho de software, desenvolvimento, testes e entrega, gestão de processos e de qualidade.

4.4.5. Syllabus:

Development and management of a software development project for a real external client; requirements, architecture, design, development, testing and deployment of a software product; project and process management and quality assurance.

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

O projeto de Estúdio é o contexto real no qual os alunos aplicam os conhecimentos-chave que adquirem nas disciplinas do ciclo de estudos e que se encontram reflectidos nos objectivos programáticos listados acima. Esta unidade curricular é portanto a componente integradora de todos os conteúdos deste ciclo de estudos, complementada pelos aspectos humanos inerentes a trabalhar em equipa.

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

The Studio project serves as the real-world context where students apply the knowledge and competences acquired in the other curricular units, as listed above. Therefore, this course is the integration backbone of this cycle of studies, complemented by the human aspects intrinsic to a team work.

4.4.7. Metodologias de ensino (avaliação incluída):

No Estúdio os alunos realizam um projeto real para uma entidade externa passando por todas as fases de desenvolvimento de software. Os projetos são realizados em equipas de sete ou mais alunos (impossibilitando a gestão informal) e têm total autonomia de gestão.

Os mentores orientam os alunos na aplicação dos métodos e técnicas referidas nas unidades curriculares e a trazer para discussão os problemas encontrados. Nas reuniões de mentoria os alunos apresentam o trabalho desenvolvido sendo questionados de forma estruturada sobre a aplicação do seu esforço e como avaliam o impacto das suas escolhas e decisões.

Toda a actividade realizada no âmbito do projeto de estúdio é armazenada numa plataforma moderna de gestão de projectos de software (git) que regista as contribuições individuais as quais são regularmente monitorizadas pelos mentores. A avaliação final de Estúdio segue as normas regulamentares de defesa de tese de mestrado.

4.4.7. Teaching methodologies (including students' assessment):

In the Studio students develop a real-world project for an external client going through the full lifecycle of software development. Work is performed in teams of more than seven students (so that informal management is impossible) and have total management autonomy.

Mentors supervise the students application of the methods and techniques acquired in the other courses of this cycle of studies and promote discussion on the difficulties encountered. In the mentoring meetings students present the work developed in the previous iteration, and are questioned in a structured way about how they are applying their effort and how do they assess the impact of their choices and decisions.

All the activities performed in the studio are logged in a typical modern software project development platform (git) which records every individual contribution. These are regularly monitored by the mentors. The final Studio assessment conforms to the legal requirements for a Master Thesis defence.

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

A melhor forma de adquirir competências de gestão de projecto de software é realizando um projecto real, não um projecto académico, pois estes não expõem os alunos às condicionantes, imprecisões e contradições inerentes à imprevisibilidade do ambiente actual do desenvolvimento de software.

Consequentemente, numa formação avançada como a proposta, não seria possível expor os alunos a este contexto onde não há dois projectos iguais, sem a execução de um projecto real na sua totalidade. Este contexto, complementado pelo acompanhamento das equipas de alunos por mentores da área de engenharia de software (docentes responsáveis pelas unidades curriculares) e com larga experiência na indústria, é o modelo pedagógico que melhor garante a partilha de conhecimentos e know-how vivenciado que complementa os conceitos adquiridos nas unidades curriculares.

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

The most effective way to acquire advanced software project management competencies is by performing a real project, not academic, as these do not expose students to the constraints, imprecisions and contradictions intrinsic to the unpredictability of contemporary software development.

Therefore, in an advanced course as the one herein proposed, it would not be possible to expose students to a reality where no two projects are alike without making them experience the full lifecycle of a real software project. This context, complemented by student mentoring by experts in software engineering (also responsible for the other curricular units), and with a strong industry experience, is the pedagic model best fit to the sharing of the know-how and experiences that complement the concepts acquired in the other curricular units.

4.4.9. Bibliografia de consulta/existência obrigatória:

A mesma bibliografia das disciplinas do ciclo de estudos (formação integrada).

The bibliography associated to the curricular units of this cycle of studies (integrated learning).

Mapa IV - Gestão do Desenvolvimento de Software

4.4.1.1. Designação da unidade curricular:

Gestão do Desenvolvimento de Software

4.4.1.1. Title of curricular unit:

Managing Software Development

4.4.1.2. Sigla da área científica em que se insere:

INF

4.4.1.3. Duração (anual, semestral ou trimestral):

Semestral / Semester

4.4.1.4. Horas de trabalho (número total de horas de trabalho):

162

4.4.1.5. Horas de contacto:

T-28; PL-14

4.4.1.6. Créditos 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):

Marco Paulo Amorim Vieira; contacto: 28h

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

José Alexandre D'Abruzzo Pereira; contacto: 14h

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

Os alunos deverão ser capazes de: escrever um plano de gestão de um projecto de software, abordando questões de análise de risco, cronograma, custos, organização da equipa, recursos e abordagem técnica; definir as áreas chave do processo e a tecnologia e as práticas associadas a cada e uma variedade de modelos de ciclos de vida do desenvolvimento de software e explicar os pontos fortes, pontos fracos, e aplicabilidade de cada um. Deverão ser capazes de compreender a relação entre produtos de software e produtos globais (se incorporados), ou o papel do produto na linha de produtos da organização. Compreender a finalidade e as limitações de padrões de desenvolvimento de software e ser capaz de aplicar adaptações sensatas quando necessário.

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

After completing this course, students will be able to: write a software project management plan, addressing issues of risk analysis, schedule, costs, team organization, resources, and technical approach; be able to define the key process areas of the Capability Maturity Model and the technology and practices associated with each and a variety of software development life cycle models and explain the strengths, weaknesses, and applicability of each. Be able to understand the relationship between software products and overall products (if embedded), or the role of the product in the organizational product line. Understand the legal issues involved in liability, warranty, patentability, and copyright. Understand the purpose and limitations of software development standards and be able to apply sensible tailoring where needed.

4.4.5. Conteúdos programáticos:

1. Conceitos Base

1.1. Engenharia de Sistemas

1.2. Ciclos de Vida

1.3. Processos de Engenharia

2. Definição de Âmbito e Gestão de Requisitos

3. Planeamento e Acompanhamento de Projetos

3.1. Estimativas de Esforço

3.2. Análise de Riscos

3.3. Planeamento

3.4. Monitorização**4. Aquisição e Contratação****4.1. Identificação de Alternativas****4.2. Modelos para Tomada de Decisão****4.3. Modelos para Avaliação Económica****5. Gestão de Expectativas de Clientes****5.1. Comunicação****5.2. Tomada de Decisão****5.3. Negociação****6. Gestão de Equipas e Pessoas****6.1. Gestão de Relacionamentos****6.2. Gestão de Conflitos****6.3. Liderança****7. Gestão da Garantia de Qualidade****4.4.5. Syllabus:****1. Base Concepts****1.1. Systems Engineering****1.2. Lifecycles****1.3. Engineering processes****2. Scope Definition and Requirements Management****3. Projects Planning and Tracking****3.1. Effort Estimation****3.2. Risk Analysis****3.3. Planning****3.4. Monitoring****4. Acquisition and Contracting****4.1. Identifying Alternatives****4.2. Models for Decision Making****4.3. Models for Cost Assessment****5. Managing Customer Expectations****5.1. Communication****5.2. Decision Making****5.3. Negotiation****6. Team and People Management****6.1. Relationship Management****6.2. Conflict Management****6.3. Leadership****7. Managing Quality Assurance****4.4.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular:**

O desenvolvimento de software de larga escala requer a capacidade de gerir os recursos tanto Humanos como computacionais através do controlo do processo de desenvolvimento. Esta unidade curricular é um curso de amplitude orientada, projetado para ajudar engenheiros de software tecnicamente treinados para adquirir o conhecimento e as competências necessárias para liderar uma equipa de projeto, entender a relação do desenvolvimento de software na engenharia geral do projeto, tempo estimado e os custos, e compreender o processo de software . A natureza do desenvolvimento de software é suficientemente única para exigir técnicas de gestão especializadas, especialmente nas áreas de estimativa e agendamento.

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

Large scale software development requires the ability to manage resources - both human and computational - through control of the development process. This course is a breadth oriented course, designed to help technically-trained software engineers to acquire the knowledge and skills necessary to lead a project team, understand the relationship of software development to overall project engineering, estimate time and costs, and understand the software process. The nature of software development is sufficiently unique to require specialized management techniques, especially in the areas of estimating and scheduling.

4.4.7. Metodologias de ensino (avaliação incluída):

Esta UC inclui aulas teóricas de exposição detalhada de conceitos, princípios e técnicas fundamentais de gestão de projetos de software, em paralelo com exemplos práticos com a intenção de ampliar o interesse dos alunos nos conceitos teóricos e exemplificar a sua aplicação em situações reais. A disciplina está organizada à volta de trabalhos práticos semanais e um conjunto de projetos em grupo. Grande parte dos trabalhos e projetos são discutidos nas aulas de modo a levar os alunos a refletir sobre as várias respostas possíveis. A avaliação inclui estas três componentes: trabalhos práticos semanais (15%), os projectos em equipa (70%) e um exame escrito final (15%).

4.4.7. Teaching methodologies (including students' assessment):

This course includes theoretical classes for detailed discussion of concepts, principles and fundamental

techniques for project management, in parallel with practical examples focusing on enlarging the interest of the students in the theoretical concepts and exemplify their applicability in real cases. The course is organized around weekly practical assignments and a set of group projects. Most assignments and projects are discussed in classes in order to lead the students to reflect on the multitude of possible solutions. Assessment includes these three components: short weekly assignments (15%), the group projects (70%) and a final written exam (15%).

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

A motivação principal para os exercícios de casa e para os projetos é dar aos alunos a oportunidade de refletir sobre as teóricas, metodologias e técnicas da unidade curricular. Os alunos são encorajados a discutir os trabalhos de casa com outros alunos, ainda que a entrega final seja única e exclusivamente o trabalho de cada um (excepto nos projetos de grupo). Os TPC estão desenhados para seguir de perto a matéria das aulas teóricas semanais, permitindo aos alunos aplicar o conhecimento obtido durante a semana imediatamente a seguir. Há projetos, desenhados para dar aos alunos a possibilidade de aplicar ideias da unidade curricular em cenários semirealistas. Cada projeto é elaborado em equipa e espera-se que os membros participem de igual forma e que seja, capazes de aplicar a matéria teórica e ligar conhecimentos distintos. Isto promove não só a consolidação dos conhecimentos teóricos, mas também a capacidade de raciocínio crítico sobre a atividade de gestão de projetos de software.

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

The purpose of the homework assignments and projects is to give students practice in using the theories, methodologies, and techniques of the course. Students are encouraged to discuss homework with other students, but the final writeup must be their own work. The homework exercises are designed to follow the weekly lectures, allowing students to apply the theoretical knowledge obtained in each lecture during the week immediately after it is lectured. There are group projects that are designed to give students a chance to apply the ideas of the course to semirealistic case studies. Each project is completed by a team. Team members are expected to participate equally in the projects. These midsize projects ensure that students are able to apply the theoretical understanding and connect different pieces of knowledge. This not only promotes a consolidation of the theory, but also provides the means for students to critically reason about the software projects management activity.

4.4.9. Bibliografia de consulta/existência obrigatória:

- 1. Tsui and Karam (T&K), "Essentials of Software Engineering." 2007, Jones and Barrlet Publishing. ISBN-13: 978-07637-3537-1*
- 2. Pressman, Roger S., 2005, "Software Engineering, A Prationer's Approach, Sixth Edition," McGraw Hill, ISBN 0-07-301933-X*

Mapa IV - Métodos: Decidir o que Construir

4.4.1.1. Designação da unidade curricular:

Métodos: Decidir o que Construir

4.4.1.1. Title of curricular unit:

Methods: Deciding what to Design

4.4.1.2. Sigla da área científica em que se insere:

INF

4.4.1.3. Duração (anual, semestral ou trimestral):

Semestral / Semester

4.4.1.4. Horas de trabalho (número total de horas de trabalho):

162

4.4.1.5. Horas de contacto:

T-28; PL-14

4.4.1.6. Créditos 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):

Mário Alberto da Costa Zenha Relá; contacto: 28h

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

Nuno Alexandre Martins Seixas; contacto: 14h

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

O desenvolvimento efetivo de software exige o domínio de diversos métodos para transpor o fosso entre o problema concreto a ser resolvido e um sistema informático funcional. Nesta unidade curricular os alunos estudam e aplicam várias técnicas para compreender e descrever o problema que estão a tentar resolver, os diversos fatores que condicionam as soluções possíveis e as diversas abordagens para escolher as melhores alternativas.

Após a frequência desta unidade curricular os estudantes estarão em condições de:

- **Identificar diferentes tipos de problemas e as respetivas estruturas;**
- **Analisar as condicionantes de natureza técnica, organizacional, de usabilidade e de mercado que influenciam a solução a adotar;**
- **Aplicar uma genuína abordagem de engenharia para solucionar o problema.**

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

The professional activity of software development requires an understanding of successful methods for bridging the gap between a problem to be solved and a working software system. In this course students will study a variety of techniques to understand the problem they're solving, the various factors that constrain the possible solutions, and approaches to deciding among alternatives.

After completing this course, students will be able to:

- **Identify different classes of problems and their structures;**
- **Analyze technical, organizational, usability, and business constraints on solutions;**
- **Apply a sound engineering approach to frame solutions.**

4.4.5. Conteúdos programáticos:

- 1.Engenharia de Requisitos;**
- 2.Design Contextual;**
- 3.Casos de Uso (como guia para o ciclo de vida do software);**
- 4.Modelação orientada aos objectivos.**
- 5.Considerações de Usabilidade;**
- 6.Condicionantes económicas, de negócio e regulamentares;**
- 7.Uma abordagem de engenharia ao desenvolvimento de software.**

4.4.5. Syllabus:

- 1.Requirements Engineering;**
- 3.Contextual Design;**
- 4.Use Cases;**
- 5. Goal-oriented modelling;**
- 5.Usability issues;**
- 6.Business, economic and policy constraints;**
- 7.An engineering approach to software development.**

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

As diversas abordagens são apresentadas sequencialmente sendo cada uma delas aplicada no projecto de Estúdio para que os alunos se vão apercebendo das características e adequação das diversas técnicas aos diferentes tipos de problemas.

As aulas presenciais são utilizadas para discutir os tópicos da unidade curricular. Esta metodologia de ensino promove a aprendizagem autónoma, bem como o desenvolvimento de competências de aprendizagem e argumentação, essenciais para preparar os alunos na capacidade de identificar e analisar problemas.

Os trabalhos de síntese relativos aos entregáveis intermédios complementam e reforçam as competências de investigação autónoma, análise crítica e competências de apresentação.

Dado que cada um dos tópicos desta u.c. são trabalhados em profundidade no Estúdio realizado em equipa, isto promove a aplicação de uma abordagem de engenharia, dado que os estudantes serão confrontados com a necessidade de compromisso entre as diversas condicionantes em jogo

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

The different approaches are presented sequentially and applied onto the Studio project so that students become aware of the different characteristics of each technique and their adequacy to the different types of problems.

Lectures are used to discuss the topics addressed. This will promote autonomous learning and the development of

discussion skills which are essential to prepare students to identify and analyze problems in their professional careers.

The synthesis work complement and reinforce both research competences, critical thinking and presentation skills. Since each topic addressed are detailed in the Studio project, developed as a team, this will promote the application of engineering approaches to frame solutions, as they will be faced with the need to balance competing constraints.

4.4.7. Metodologias de ensino (avaliação incluída):

O processo de aprendizagem envolve a aplicação concreta das abordagens e a sua apresentação à turma, mais do que ouvir exposições teóricas. Assim, esta UC inclui aulas teóricas de exposição de conceitos, em paralelo com exemplos práticos com a intenção de ampliar o interesse dos alunos nos conceitos teóricos e exemplificar a sua aplicação em situações reais.

A disciplina está organizada em torno do projecto de Studio através da entrega faseada de (sete) artefactos de aplicação de cada uma das técnicas ao projecto, os quais são apresentados, discutidos e dado feedback, quer por escrito, quer em sala de aula.

A avaliação inclui os seguintes componentes:

- (seis) artefactos intermédios, perfazendo 75%.
- Documento de requisitos final consolidado: 25%.

4.4.7. Teaching methodologies (including students' assessment):

The learning process involves applying the ideas and explaining them to the peers, rather than by listening to lectures. Therefore, this course includes lectures with the exposition of concepts, along with practical examples intended to increase students' interest in theoretical concepts and exemplify their application to real situations. The course contents are organized around the Studio project by the phased delivery of (seven) artifacts where the different techniques are applied, which are presented, discussed, and feedback is provided both in writing and in class. The grading includes the following components:

- (six) intermediate artifacts: 75%
- A final consolidated Requirements document: 25%.

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

Um artigo descrevendo em detalhe a metodologia desta cadeira pode ser encontrado em:

<http://www.casos.cs.cmu.edu/publications/papers/2005DecidingWhatDesign.pdf>

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

A paper describing in detail the rationale for this course can be found at:

<http://www.casos.cs.cmu.edu/publications/papers/2005DecidingWhatDesign.pdf>

4.4.9. Bibliografia de consulta/existência obrigatória:

1. [BRO95] *The Mythical Man-Month*, Fred Brooks, ISBN 0201835959, 336 pp., Addison-Wesley Professional; 2ª ed. 1995.
2. [BEY16] *Hugh Beyer and Karen Holtzblat: Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufman, 2016. ISBN13: 978-0128008942
3. [ArMi01] *Frank Armour and Granville Miller: Advanced Use Case Modeling: Software Systems*. Addison-Wesley, 2001. ISBN-13: 978-0201615920
4. [COOP14] *About Face: The Essentials of Interaction Design*, Alan Cooper, Robert Reimann, David Cronin, Christoffer Noessel, Wiley, 4th edition (2014), 720 pp., ISBN-13 : 978-1118766576
5. [NOR13] *The Design of Everyday Things*, 2ed. , Donald Norman, ISBN 0262525674, 368 pp., MIT Press, 2013.

Mapa IV - Modelos de Sistemas de Software

4.4.1.1. Designação da unidade curricular:

Modelos de Sistemas de Software

4.4.1.1. Title of curricular unit:

Models of Software Systems

4.4.1.2. Sigla da área científica em que se insere:

INF

4.4.1.3. Duração (anual, semestral ou trimestral):

Semestral / Semester

4.4.1.4. Horas de trabalho (número total de horas de trabalho):

4.4.1.5. Horas de contacto:

T-28; PL-14

4.4.1.6. Créditos 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):

Raul André Brajczewski Barbosa; contacto: 28h

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

Nuno Alexandre Martins Seixas; contacto: 14h

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

Os fundamentos científicos da Engenharia de Software dependem do uso de modelos abstratos e lógicas com elevada precisão, por forma a caracterizar e raciocinar sobre propriedades de sistemas de software. Existem diversos modelos básicos e lógicas que, ao longo do tempo, provaram ser particularmente importantes e ubíquas no estudo de sistemas de software. Esta unidade curricular foca esse corpo de conhecimento.

Após a conclusão desta disciplina, os alunos conseguirão (i) compreender as forças e fraquezas de certos modelos e lógicas, incluindo máquinas de estados, modelos algébricos e de processos, e lógica temporal, (ii) selecionar e descrever modelos abstratos apropriados para diversas classes de sistemas, descrever relações de abstração entre diferentes níveis de descrição, e raciocinar acerca da correção de refinamentos, e (iii) provar propriedades elementares de sistemas modelados através de técnicas introduzidas pela unidade curricular.

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

Scientific foundations for Software Engineering depend on the use of precise, abstract models and logics for characterizing and reasoning about properties of software systems. There are a number of basic models and logics that over time have proven to be particularly important and pervasive in the study of software systems. This course is concerned with that body of knowledge.

After completing this course, students will (i) understand the strengths and weaknesses of certain models and logics including state machines, algebraic and process models, and temporal logic, (ii) be able to select and describe appropriate abstract formal models for certain classes of systems, describe abstraction relations between different levels of description, and reason about the correctness of refinements, and (iii) prove elementary properties of modeled systems using techniques introduced in the course.

4.4.5. Conteúdos programáticos:

- *Introdução: informação sobre a disciplina; o que um modelo é.*
- *Fundamentos: lógica; técnicas de prova; sequências; conjuntos, relações e funções.*
- *Sistemas de eventos discretos; máquinas de estados finitos; raciocínio sobre máquinas de estados.*
- *Introdução à verificação de modelos; verificação de programas sequenciais.*
- *Concorrência; máquinas de estados concorrentes; técnicas de modelação; raciocínio sobre concorrência.*
- *Lógica Temporal Linear; propriedades de segurança e vivacidade; raciocínio automático e verificação formal.*
- *Modelação de sistemas distribuídos; raciocínio sobre processos comunicantes.*
- *Métodos formais no mundo real.*

4.4.5. Syllabus:

- *Introduction: course information; what a model is.*
- *Foundations: logic; proof techniques; sequences; sets, relations, and functions.*
- *Discrete event systems; finite-state machines; reasoning about state machines.*
- *Introduction to model checking; verification of sequential programs.*
- *Concurrency; concurrent state machines; modeling techniques; reasoning about concurrency.*
- *Linear Temporal Logic; safety and liveness properties; automated reasoning and formal verification.*
- *Modeling distributed systems; reasoning about communicating processes.*
- *Formal methods in the real world.*

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

curricular:

Esta UC considera muitos dos modelos comuns para representar sistemas sequenciais e concorrentes, tais como máquinas de estados e álgebras. Mostra como lógicas distintas podem ser usadas para especificar propriedades de sistemas de software, como correção funcional, ausência de deadlock, e consistência interna. Mecanismos de composição, relações de abstração, invariantes, não-determinismo, e descrições indutivas. A UC avança para as matérias de máquinas de estados, processos de estados finitos, e para as correspondentes técnicas de raciocínio. A linguagem formal Promela é introduzida como uma forma de especificar formalmente sistemas computacionais. Os alunos estarão em condições de provar propriedades elementares de sistemas descritos nos modelos formais introduzidos. A Lógica Temporal Linear é introduzida como um meio para raciocinar acerca de sistemas concorrentes e distribuídos.

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

MSS considers many of the standard models for representing sequential and concurrent systems, such as state machines, algebras and traces. It shows how different logics can be used to specify properties of software systems, such as functional correctness, deadlock freedom, and internal consistency. Composition mechanisms, abstraction relations, invariants, non-determinism, and inductive and notational descriptions are themes throughout the course. The course focuses on state machines, finite state processes, and the related reasoning techniques. The Promela formal language is then introduced as the means to formally specify computer systems. Here, students should be able to prove elementary properties of systems described by the introduced models. Students should be able to select and describe appropriate formal models. Linear Temporal Logic is presented as a means for reasoning about distributed and concurrent systems.

4.4.7. Metodologias de ensino (avaliação incluída):

Esta unidade curricular inclui aulas teóricas de exposição detalhada de conceitos, princípios e técnicas fundamentais de modelação formal, em paralelo com exemplos práticos para estimular o interesse dos alunos pelos conceitos teóricos e exemplificar a sua aplicação a situações reais. Para dar mais tempo para responder a questões detalhadas há aulas semanais de natureza prática e de discussão. Estas sessões têm como objetivo olhar de perto para os trabalhos de casa semanais e para os projetos. Os alunos têm leituras semanais, com exercícios para resolver em casa, e três projetos em grupo. Projectos de aplicação: 70%, Exame final: 30%

4.4.7. Teaching methodologies (including students' assessment):

This course includes theoretical lectures with detailed exposition of concepts, principles and fundamental techniques of formal modelling, along with practical examples to increase students' interest in theoretical concepts and exemplify their application to real situations. To allow for more time to answer questions there are weekly "recitation" sessions. These sessions are practical in nature and suitable to take a closer look at the homework exercises and the projects. The course is organized around weekly reading assignments, with homework exercises, and three group projects. Application projects: 70%; Final Exam: 30%

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

A motivação principal para as leituras com exercícios para casa, assim como para os projetos, é dar aos alunos a oportunidade de usar os modelos, lógicas e ferramentas da unidade curricular. Por forma a dar aos alunos o máximo de oportunidades para aprender a partir dos projetos e dos trabalhos de casa, é permitido refazer problemas que não recebam aprovação. Há três projetos, cada um elaborado em equipa, no qual os membros devem participar de igual forma. Uma revisão por pares é distribuída no final do semestre. Estes projetos de média dimensão asseguram que os alunos são capazes de aplicar a matéria teórica e ligar conhecimentos distintos, por forma a selecionar e descrever modelos formais abstratos para certas classes de sistemas. Isto promove não só a consolidação dos conhecimentos teóricos, mas também a capacidade de raciocínio crítico sobre as propriedades elementares dos sistemas modelados.

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

The main motivation for reading assignments with home exercises, as well as for projects, is to give students the opportunity to use the models, logics and tools of the course. In order to give students the maximum opportunity to learn from the projects and the homework it is permitted to redo problems that do not receive approval. There are three designs, each prepared in a team whose members must participate equally. A peer review is distributed at the end of the semester. These average size projects ensure that students are able to apply the theoretical material and connect distinct knowledge, in order to select and describe abstract formal models for certain classes of systems. This promotes not only the consolidation of theoretical knowledge, but also the critical thinking skills on the elementary properties of modeled systems.

4.4.9. Bibliografia de consulta/existência obrigatória:

Livros essenciais | Required textbooks:

- *Introduction to Discrete Event Systems*, C. G. Cassandras et al., Springer, 2008.
- *Principles of the Spin Model Checker*, M. Ben-Ari, Springer, 2008.

Opcional | Optional:

- *Logic in Computer Science: Modelling and Reasoning about Systems*, M. Huth et al., Cambridge, 2004.

- *Concurrency: State Models and Java Programs*, J. Magee et al., 2nd Ed., Wiley, 2006.
- *The Spin Model Checker: Primer and Reference Manual*, G. J. Holzmann, Addison-Wesley, 2004.
- *Verificação de Modelos*, R. Barbosa, 2018.

Suplementar | Supplemental:

- *Concepts and Notations for Concurrent Programming*, G. R. Andrews et al., *Computing Surveys*, Vol. 15, No. 1, 1983.
- *Formal Methods: State of the Art and Future Directions*, E. M. Clarke et al., *ACM Computing Surveys*, Vol. 28, No. 4, 1996.
- *A Primer on Model Checking*, M. Ben-Ari, *ACM Inroads*, Vol. 1, No. 1, 2010.
- *The Model Checker Spin*, G. J. Holzmann, *IEEE Transaction on Software Engineering*, Vol. 23, No. 5, 1997.

Mapa IV - Análise de Artefactos de Software

4.4.1.1. Designação da unidade curricular:

Análise de Artefactos de Software

4.4.1.1. Title of curricular unit:

Analysis of Software Artifacts

4.4.1.2. Sigla da área científica em que se insere:

INF

4.4.1.3. Duração (anual, semestral ou trimestral):

Semestral / Semester

4.4.1.4. Horas de trabalho (número total de horas de trabalho):

162

4.4.1.5. Horas de contacto:

T-28; PL-14

4.4.1.6. Créditos 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):

Henrique dos Santos Carmo Madeira; contacto: 28h

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

Nuno Alexandre Martins Seixas; contacto: 14h

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

Esta UC foca a análise de artefatos de software de código principalmente, mas também incluindo a análise de projetos, arquiteturas e testes. Centrando-se nos aspetos de qualidade e características não funcionais, abordada técnicas de análise estática e dinâmica, tais como análise estática código, inspeções, verificação de modelos, diferentes estratégias de testes e benchmarking. Após a conclusão desta uc, os alunos deverão: saber que tipos de análises estão disponíveis e como usá-las, compreender o seu alcance e poder, quando podem ser aplicadas, que conclusões podem ser tiradas a partir dos resultados; ter noções fundamentais suficientes para avaliar novos tipos de análise, quando as mesmas forem desenvolvidas; ter alguma experiência em selecionar e escrever análises para uma verdadeira peça de software, aplicando-as e interpretando os resultados.

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

This course focuses on analysis of software artifacts primarily code, but also including analysis of designs, architectures, and test suites. The course is centered on software quality and non-functional aspects, addressing static and dynamic techniques, such as static code analysis, inspections, model checking, different test strategies and benchmarking. The course balances theoretical discussions with lab exercises in which students apply the ideas they are learning to real artifacts. After completing this course, students will: know what kinds of analyses are

available and how to use them; understand their scope and power, when they can be applied, and what conclusions can be drawn from their results; have a grasp of fundamental notions sufficient to evaluate new kinds of analysis when they are developed; have some experience in selecting and writing analyzes for a real piece of software, applying them and interpreting the results.

4.4.5. Conteúdos programáticos:

Introdução à qualidade e verificação; Inspeções; modelagem de teste de domínio; verificação de modelos; critérios de adequação, testes caixa preta (particionamento em classes de equivalência; análise de valores fronteira), testes caixa branca (controlo de fluxo; fluxo de dados) teste Combinatório; testes aleatórios e mutações, análise de Arquitetura.

4.4.5. Syllabus:

Introduction to quality and verification; Inspections; Test domain modeling; model checking; blackbox testing (equivalence class partitioning; boundary values analysis), withbox testing (control flow; data flow), combinatorial testing; Framework; Mutation testing; Analysis; Formal verification; Architectural analysis.

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

A capacidade de construir, manter e reutilizar sistemas de software depende da capacidade de efetivamente analisar os produtos de desenvolvimento de software. Esta UC foca-se nos vários tipos de artefatos de software e apresenta métodos para analisá-los. É dada ênfase especial à análise de funcionalidade para descobrir defeitos em artefatos e para apoiar a manutenção e engenharia reversa. A disciplina apresenta vários tipos de análise para tratar outros tipos de sistemas de propriedades, tais como desempenho e segurança. Considera as semelhanças fundamentais entre os métodos de análise para destacar limitações e âmbitos e equilibra as discussões teóricas com exercícios de laboratório em que os alunos aplicam as ideias que aprendem a artefatos reais.

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

The ability to build, maintain and reuse software systems depends on the capacity to effectively analyze the products of software development. This course focuses on various kinds of software artifacts and introduces methods for analyzing them. Special emphasis is given to the analysis of functionality for discovering defects in artifacts and to support maintenance and reverse engineering. The course introduces various kinds of analysis to address other kinds of systems properties, such as performance and security. It considers the fundamental similarities among analysis methods to highlight limitations and scopes, and balances theoretical discussions with lab exercises in which students apply the ideas they are learning to real artifacts.

4.4.7. Metodologias de ensino (avaliação incluída):

Na prática, esta UC inclui aulas teóricas de exposição detalhada de conceitos, princípios e técnicas fundamentais de análise de artefactos de software, em paralelo com exemplos práticos com a intenção de ampliar o interesse dos alunos nos conceitos teóricos e exemplificar a sua aplicação em situações reais. A avaliação inclui testes, apresentações de projeto, exame, trabalhos de casa.

4.4.7. Teaching methodologies (including students' assessment):

In practice, this course includes theoretical lectures with detailed exposition of concepts, principles and fundamental techniques for analyzing software artifacts, along with practical examples intended to increase students' interest in theoretical concepts and exemplify their application to real situations. The evaluation includes tests; project presentations; exam, and homework assignments.

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

A motivação principal para os exercícios de casa e para os projetos é dar aos alunos a oportunidade de refletir sobre as teóricas, metodologias e técnicas da unidade curricular. Os alunos são encorajados a discutir os trabalhos de casa com outros alunos, ainda que a entrega final seja única e exclusivamente o trabalho de cada um.

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

The purpose of the homework assignments and projects is to give students practice in using the theories, methodologies, and techniques of the course. Students are encouraged to discuss homework with other students, but the final write-up must be their own work.

4.4.9. Bibliografia de consulta/existência obrigatória:

*Design Patterns Explained;
Buechner, Test Case Design Using the Classification Tree
Method; Conrad, A Systematic Approach to Testing Automotive Control Software*

Mapa IV - Arquiteturas para Sistemas de Software

4.4.1.1. Designação da unidade curricular:
Arquiteturas para Sistemas de Software

4.4.1.1. Title of curricular unit:
Architectures for Software Systems

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

4.4.1.3. Duração (anual, semestral ou trimestral):
Semestral / Semester

4.4.1.4. Horas de trabalho (número total de horas de trabalho):
162

4.4.1.5. Horas de contacto:
T-28; PL-14

4.4.1.6. Créditos 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):
Bruno Miguel Brás Cabral; contacto: 28h

4.4.3. Outros docentes e respetivas cargas letivas na unidade curricular:
António Pedro Gaspar Damasceno; contacto: 14h

4.4.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):
Após a conclusão desta unidade curricular, os alunos deverão: ser capazes de descrever uma arquitetura com precisão; reconhecer os principais estilos arquiteturais existentes em sistemas de software; propor alternativas de arquitetura para um problema, e escolher entre eles; ter as competências necessárias para a construção de um sistema de software de dimensão média que satisfaça uma especificação de arquitetura; usar as ferramentas de desenvolvimento corretas para acelerar essas tarefas; compreender a definição formal de uma arquitetura e ser capaz de raciocinar sobre as suas propriedades; usar o conhecimento de domínio para especializar uma arquitetura para uma determinada família de aplicações.

4.4.4. Intended learning outcomes (knowledge, skills and competences to be developed by the students):
After completing this course, students will: be able to describe an architecture accurately; capably recognize major architectural styles in existing software systems; generate architectural alternatives for a problem, and choose among them; have the skills needed to construct a medium sized software system that satisfies an architectural specification; use existing definitions and development tools to expedite such tasks; understand the formal definition of a number of architectures and be able to reason about the properties of those architectures; use domain knowledge to specialize an architecture for a particular family of applications.

4.4.5. Conteúdos programáticos:
Conceitos base de Arquitetura de Software (AS): o papel da AS no desenvolvimento de software
Principais estilos e padrões arquiteturais em software
Análise de requisitos de software e o seu papel na construção e avaliação de uma AS
Desenho de uma AS: criação, fundamentação e seleção de alternativas arquiteturais
Reutilização de software: estratégias para reutilização de software e avaliação do seu impacto na AS. O recurso a COTS e a middleware numa AS
Notações formais para a especificação e análise de AS
Documentação de uma AS
Implementação de uma AS: design detalhado e design patterns
Avaliação de uma AS: respeito pelos requisitos do software e a gestão de compromissos entre atributos de qualidade
Consideração de tendências futuras do software e o seu impacto nas AS

4.4.5. Syllabus:

Fundamental concepts of Software Architecture (SA): the role of SA in software development
Major architectural styles and patterns in software
Analysis of software requirements and its role in the construction and evaluation of an AS
Drawing an AS: creation, reasoning and selection of architectural alternatives
Reusing software: strategies for software reuse and assessing its impact on SA. The use of COTS middleware and an SA
Formal notations for specifying and analyzing AS
Documentation of a SA
Implementation of a SA: detailed design and design patterns
Evaluation of a SA: respect for requirements management and software tradeoffs between quality attributes
Consideration of future trends in the software and its impact on SA

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

O projeto de um sistema de software complexo exige dos seus criadores a capacidade de descrever, avaliar e criar sistemas a um nível de abstração arquitetural. Esta disciplina introduz o projeto arquitectónico de sistemas de software complexos e descreve os processos implícitos ao seu desenvolvimento. São descritas as estruturas de software mais utilizadas, técnicas para a concepção e implementação dessas estruturas, modelos e notações formais para caracterizar e raciocinar sobre arquiteturas, ferramentas para a geração de instâncias específicas de uma arquitetura e estudos de caso de arquiteturas de sistemas reais. Esta unidade curricular fornece aos alunos as competências necessárias para saber avaliar arquiteturas de sistemas existentes e projetar novos sistemas de forma fundamentada através da utilização correta de paradigmas arquitectónicos convencionados.

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

Successful design of complex software systems requires the ability to describe, evaluate, and create systems at an architectural level of abstraction. This course introduces architectural design of complex software systems. The course considers commonly used software system structures, techniques for designing and implementing these structures, models and formal notations for characterizing and reasoning about architectures, tools for generating specific instances of an architecture, and case studies of actual system architectures. It teaches the skills and background students need to evaluate the architectures of existing systems and to design new systems in principled ways using well-founded architectural paradigms.

4.4.7. Metodologias de ensino (avaliação incluída):

Aulas teóricas com exposição detalhada, recorrendo a meios audiovisuais, dos conceitos, princípios e teorias fundamentais sobre a criação e avaliação de arquiteturas de sistemas de software. Nestas aulas também se pretende que os alunos adquiram conhecimentos sobre ferramentas e técnicas de desenvolvimento e avaliação de arquiteturas de software e que, com a orientação do docente, desenvolvam a arquitetura do seu projeto de Estúdio de Desenvolvimento de Software (outra UC do curso), assim como executem e discutam a realização de 3 trabalhos práticos.
Projectos de aplicação: 45%; Resolução de problemas: 15%; Exame final: 40%

4.4.7. Teaching methodologies (including students' assessment):

Theoretical classes with detailed exposition, using visual aids, of the concepts, principles and fundamental theories for creating and assessing software architectures. In these classes students are also required to acquire knowledge about tools and techniques for developing and assessing software architectures and that, under the guidance of the teachers, develop architecture for their Software Development Studio (other unit of the program) project, as well as execute and discuss the realization of 3 practical assignments.
Application projects: 45%; Problem resolving reports: 15%; Final Exam: 40%

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

A estratégia e o método de ensino adotado procuram envolver os alunos no processo de aprendizagem e na sua valorização pessoal, e assim levar ao desenvolvimento, para além de competências técnicas específicas, de algumas competências genéricas, de natureza instrumental, pessoal e sistémicas.

Com os exemplos e exercícios práticos apresentados nas aulas teóricas estão criadas as condições para o desenvolvimento das competências para análise e síntese, resolver problemas, decisão, raciocínio crítico, aprendizagem autónoma, adaptabilidade a novas situações, e em aplicar na prática os conhecimentos teóricos adquiridos nas aulas teóricas e práticas-laboratoriais. Os trabalhos práticos permitem aos alunos, explorar e exercitar pela primeira vez os temas abordados nas teóricas, obter feedback do seu desempenho, antes de os aplicarem ao projeto de Estúdio de Desenvolvimento de Software. Com o projeto que os alunos desenvolvem, são criadas as condições para o desenvolvimento das competências

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

The teaching strategy and methods adopted aim at engaging the student in the learning process and his personal development, and lead to the development of some generic competencies of instrumental, personal and systemic nature.

The practical examples and exercises presented in theoretical classes create conditions for the development of competencies for analysis and synthesis, problem solving, decision-making, critical thinking, independent learning, adaptability to new situations, and practical application of theoretical knowledge acquired both in the theoretical and practical-laboratorial classes. The practical assignments allow students to explore and practice for the first time the concepts learned on the theoretical classes, get feedback on their performance, before applying them on the project of Software Development Studio. The students project creates conditions for the development of competencies.

4.4.9. Bibliografia de consulta/existência obrigatória:

Software Architecture in Practice, Fourth Edition, by Len Bass, Paul Clements, and Rick Kazman. Addison-Wesley 2021, ISBN-13: 978-0136886099.

Documenting Software Architectures: Views and Beyond, Second Edition, by Clements, et al.. AddisonWesley 2011.

Design Patterns: Elements of Reusable Object-Oriented Software, by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Addison-Wesley, 1988.

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:

A UC garante o alinhamento na definição das Fichas de Unidade Curricular, de forma que os objetivos de aprendizagem, competências, métodos de ensino e avaliação sejam coerentes. O Conselho Científico analisa e valida as FUC e o Conselho Pedagógico analisa e discute estas matérias. Procurou-se ainda garantir a promoção desta adequação através da análise dos resultados dos inquéritos pedagógicos e definição de ações de melhoria, quando aplicável – estes inquéritos avaliam a perceção dos/as estudantes sobre os resultados da aprendizagem alcançados. Adicionalmente, ainda no âmbito dos inquéritos, os comentários dos/as estudantes e docentes são analisados e classificados, permitindo a identificação de aspetos a ajustar nas metodologias de ensino e aprendizagem e sua adequação aos objetivos de aprendizagem definidos. Esta informação é utilizada pela Coordenação do C.E. e Direção da UO, para definir e implementar melhorias.

4.5.1. Evidence of the teaching and learning methodologies coherence with the intended learning outcomes of the study programme:

The UC guarantees the alignment in the definition of the Course Unit Files (CUF) so that the learning outcomes, skills, teaching methods and evaluation are coherent. The Scientific Council analyzes and validates the CUF and the Pedagogical Council analyzes and discusses these matters. It was also sought to ensure the promotion of this adequacy by analyzing the results of the pedagogical surveys and defining improvement actions, when applicable - these surveys assess the students' perception of the learning outcomes achieved and the overall average appraisal of the learning is requested. Additionally, still in the scope of the surveys, the comments of the students and teachers are analyzed and classified, allowing the identification of aspects to be adjusted in teaching and learning methodologies and their adequacy to the defined learning outcomes. This information is used by the Coordination of the Study Programme and the Direction of the Faculty to define and implement improvements.

4.5.2. Forma de verificação de que a carga média de trabalho que será necessária aos estudantes corresponde ao estimado em créditos ECTS:

A UC procura, desde logo, garantir esta verificação através da análise dos inquéritos pedagógicos a outros ciclos de estudo com unidades curriculares análogas, sendo solicitado a estudantes e docentes que avaliem a adequação da carga de esforço exigida (ligeira, adequada, moderadamente pesada ou excessiva). Ademais, existe uma avaliação qualitativa com base nos resultados às questões dos inquéritos pedagógicos por estudantes e docentes, os quais são analisados e permitem identificar situações de possível desadequação da carga lectiva e de trabalho. Finalmente, sendo o modelo pedagógico deste ciclo de estudos baseado em projecto, faz intrinsecamente parte da gestão de projecto monitorizar e analisar continuamente o esforço aplicado.

4.5.2. Means to verify that the required students' average workload corresponds the estimated in ECTS credits:

The UC seeks, first and foremost, to guarantee this verification through the application of the pedagogical survey results of other study cycles with similar courses, where students and teachers are asked to assess the adequacy of the required workload (whether if it was low, adequate, moderately heavy or excessive). A posteriori it is also performed a qualitative analysis based on the comments submitted by students and teachers in the pedagogical surveys, which allows identifying and acting in situations of possible inadequacy of the workload. Last but not least, since this cycle of studies adopts a project-based pedagogical model, it is intrinsic to the course activity to monitor and analyze continuously the effort involved.

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/As docentes definem a avaliação de acordo com os objetivos de aprendizagem das unidades curriculares que coordenam, considerando os objetivos gerais do curso. Estes aspetos, bem como a adequação da avaliação aos

objetivos encontram-se definidos na ficha da unidade curricular, que é analisada e validada pelo Conselho Científico. A verificação desta coerência é feita em reuniões com o corpo docente e discente e reuniões do Conselho Pedagógico, permitindo a identificação de aspetos a ajustar nas metodologias de avaliação e a sua adequação aos objetivos de aprendizagem definidos.

4.5.3. Means of ensuring that the students assessment methodologies are adequate to the intended learning outcomes: *The UC guarantees the alignment in the definition of the Course Unit Files (CUF) so that the learning outcomes, skills, teaching methods and evaluation are coherent. The Scientific Council analyzes and validates the CUF and the Pedagogical Council analyzes and discusses these matters. It was also sought to ensure the promotion of this adequacy by analyzing the results of the pedagogical surveys and defining improvement actions, when applicable - these surveys assess the students' perception of the learning outcomes achieved and the overall average appraisal of the learning is requested. Additionally, still in the scope of the surveys, the comments of the students and teachers are analyzed and classified, allowing the identification of aspects to be adjusted in teaching and learning methodologies and their adequacy to the defined learning outcomes. This information is used by the Coordination of the Study Programme and the Direction of the Faculty to define and implement improvements.*

4.5.4. Metodologias de ensino previstas com vista a facilitar a participação dos estudantes em atividades científicas (quando aplicável):

Sendo este ciclo de estudos destinados a formar líderes na indústria de software, a participação em actividades científicas não é um objectivo primeiro. Contudo, acreditamos que a capacidade de análise crítica e assente em factos demonstráveis objectivamente é comum às actividades de engenharia e de investigação. Como tal, os projectos de Studio são acompanhados por investigadores do CISUC que neles podem aplicar e/ou avaliar algumas novas técnicas e metodologias de engenharia de software, quer ao nível de análise de artefactos, quer de diagnóstico de processos. Estas actividades envolvem directamente os alunos. Como resultado, não só temos alunos do actual MSE a prosseguirem os seus estudos para doutoramento em empresas proponentes de projectos de Studio (e.g. Feedzai) como temos inclusivamente trabalhos de doutoramento a decorrer que têm como case-studies projectos do actual Studio.

4.5.4. Teaching methodologies that promote the participation of students in scientific activities (as applicable):

This new cycle of studies seeks to prepare technical leaders, so the participation on research activities is not a priority focus. However, we do believe that an evidence-based approach with a critical mindset is shared with scientific research. Therefore, the Studio projects are followed by CISUC researchers, namely of the Software and Systems Engineering group, and used as testbeds for application or validation of novel software engineering techniques and methodologies. These activities involve directly the students. As a result, not only some students have enrolled into our doctoral program with scholarships sponsored by Studio project partner companies (e.g. Feedzai), but we have also active PhD students that use as case-studies current Studio 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-74/2006, na redação dada pelo DL-65/2018:

Para a criação deste mestrado foram auscultadas as entidades empregadoras deste sector; estas são diretamente envolvidas na elaboração, proposta e apoio durante os trabalhos de projeto; o seu objectivo primeiro é o desenvolvimento de competências técnicas relevantes para a indústria de software; as condições de ingresso estão alinhadas com os requisitos de uma oferta de formação ao longo da vida.

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:

During the development of this master program representatives of the national software industry were consulted; major national players are involved in the proposal, design and support of the project activities; its main goal is to develop relevant technical competences to the software industry; the application is aligned with lifelong learning constraints.

4.6.2. Forma como os docentes foram consultados sobre a metodologia de cálculo do número de créditos ECTS das unidades curriculares:

Esta proposta de evolução do actual mestrado em engenharia de software foi realizada de forma colaborativa pela equipa de docentes envolvidos. Como tal, todos estão alinhados com o cálculo do número de ECTS das unidades curriculares. Ademais, com excepção da unidade de Projecto de Studio de Software que foi redesenhado, todas as unidades curriculares são preexistentes com ajustes nos 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 current proposal for the creation of a new cycle of studies is actually an update of the current edition of the Master in Software Engineering for working professionals. This proposal was designed in a collaborative fashion by the whole team of teaching staff involved. Therefore, everyone is aligned with the number of ECTS defined for each

curricular unit. Moreover, with the exception of Software Project Studio, that was updated, all curricular units are pre-existing with ECTS adjustments..

4.7. Observações

4.7. Observações:

Este mestrado está fortemente ligado à indústria. Um grupo de empresas atuam com parceiros do MSE, e estão motivadas para melhorar o estado da prática da engenharia de software, fornecendo suporte aos alunos (suporte financeiro durante o curso, propostas de projeto para realização do Studio, etc.) e permitindo-lhes manter os seus empregos em simultâneo com a frequência do curso. Estas empresas constituem a designada 'MSE Industry Alliance' e representam um movimento que tem como objectivo modernizar as práticas da indústria nacional de software. O projeto de Studio, o elemento agregador de toda formação, é um projeto de desenvolvimento real em que uma das empresas parceiras serve de cliente à equipa de alunos alocada ao projeto. No que diz respeito à obtenção do grau de mestre, as unidades curriculares integradas com o Studio são o equivalente, ou excedem, um projeto de tese/dissertação existente nos mestrados da FCTUC.

4.7. Observations:

This master's degree is tightly linked to the national software industry. A group of companies work as partners of the program, that are committed to improve the state of the art of software engineering, and provide support to students (financial support during the course, projects to be realized within the Studio, etc.) by allowing them to maintain their jobs while attending the program. These companies are part of the 'MSE Industry Alliance' that represent a movement that aims to improve the current software industry practices. The study project, the aggregating element of all course contents, is a real development project in which one of the partner companies serves as a client to the student team assigned to the project. With regard to obtaining the master's degree, the integration of the curricular units with the Studio is equivalent to, or exceed, a thesis/dissertation project of the current master's degrees of the FCTUC.

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.

Mário Alberto da Costa Zenha Relá

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	Vínculo/ Link	Especialista / Specialist	Área científica / Scientific Area	Regime de tempo / Employment regime	Informação/ Information
Mário Alberto da Costa Zenha Relá	Professor Auxiliar ou equivalente	Doutor	Docente de Carreira (Art. 3º, alínea k) do DL-74/2006, na redação fixada pelo DL-65/2018)		Engenharia Electrotécnica	100	Ficha submetida
João Gabriel Monteiro de Carvalho e Silva	Professor Catedrático ou equivalente	Doutor	Docente de Carreira (Art. 3º, alínea k) do DL-74/2006, na redação fixada pelo DL-65/2018)		Engenharia Electrotécnica - Informática	100	Ficha submetida
Bruno Miguel Brás Cabral	Professor Auxiliar ou equivalente	Doutor	Docente de Carreira (Art. 3º, alínea k) do DL-74/2006, na redação fixada pelo DL-65/2018)		Engenharia Informática	100	Ficha submetida
Marco Paulo Amorim Vieira	Professor Catedrático ou equivalente	Doutor	Docente de Carreira (Art. 3º, alínea k) do DL-74/2006, na redação fixada pelo DL-65/2018)		Engenharia Informática	100	Ficha submetida
Raul André Brajczewski Barbosa	Professor Auxiliar ou equivalente	Doutor	Docente de Carreira (Art. 3º, alínea k) do DL-74/2006, na redação fixada pelo DL-65/2018)		Engenharia Informática	100	Ficha submetida

Henrique Santos do Carmo Madeira	Professor Catedrático ou equivalente	Doutor	Docente de Carreira (Art. 3º, alínea k) do DL-74/2006, na redação fixada pelo DL-65/2018)	Engenharia Informática	100	Ficha submetida
José Alexandre D'Abruzzo Pereira	Assistente convidado ou equivalente	Mestre	Outro	Engenharia Informática	30	Ficha submetida
Nuno Alexandre Martins Seixas	Professor Auxiliar convidado ou equivalente	Mestre	Outro	Engenharia de Software	100	Ficha submetida
António Pedro Gaspar Damasceno	Assistente convidado ou equivalente	Mestre	Outro	Engenharia de Software	30	Ficha submetida
					760	

<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.

9

5.4.1.2. Número total de ETI.

7.6

5.4.2. Corpo docente próprio – docentes do ciclo de estudos integrados na carreira docente ou de investigação (art.º 3 DL-74/2006, na redação fixada pelo DL-65/2018).

5.4.2. Corpo docente próprio – docentes do ciclo de estudos integrados na carreira docente ou de investigação (art.º 3 DL-74/2006, na redação fixada pelo DL-65/2018).* / "Career teaching staff" – teachers of the study programme integrated in the teaching or research career.*

Vínculo com a IES / Link with HEI	% em relação ao total de ETI / % of the total of FTE	
Docente de Carreira (Art. 3º, alínea k) do DL-74/2006, na redação fixada pelo DL-65/2018)	78.947368421053	100
Outro	21.052631578947	30

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):	6	78.947368421053

5.4.4. Corpo docente especializado

5.4.4. Corpo docente especializado / Specialised teaching staff.

Corpo docente especializado / Specialized teaching staff	ETI / FTE	Percentagem* / Percentage*
Doutorados especializados na(s) área(s) fundamental(is) do CE (% total ETI) / PhDs specialised in the fundamental area(s) of the study programme (% total FTE)	6	78.947368421053
Não doutorados, especializados nas áreas fundamentais do CE (% total ETI) / Staff specialised in the fundamental areas of the study programme not holding PhDs in these areas (% total FTE)	1.6	21.052631578947

Não doutorados na(s) área(s) fundamental(is) do CE, com Título de Especialista (DL 206/2009) nesta(s) área(s) (% total ETI) / Specialists not holding a PhD, but with a Specialist Title (DL 206/2009) in the fundamental area(s) of the study programme (% total FTE)	0	0
% do corpo docente especializado na(s) área(s) fundamental(is) (% total ETI)		100
% do corpo docente doutorado especializado na(s) área(s) fundamental(is) (% docentes especializados)		78.947368421053

5.4.5. Corpo Docente integrado em Unidades de Investigação da Instituição, suas subsidiárias ou polos nela integrados (art.º 29.º DL-74/2006, na redação fixada pelo DL-65/2018)

5.4.5. Corpo Docente integrado em Unidades de Investigação da Instituição, suas subsidiárias ou polos nela integrados (art.º 29.º DL-74/2006, na redação fixada pelo DL-65/2018) / Teaching Staff integrated in Research Units of the Institution, its subsidiaries or integrated centers (article 29, DL no. 74/2006, as written in the DL no. 65/2018)

Descrição	ETI / FTE	Percentagem* / Percentage*
Corpo Docente integrado em Unidades de Investigação da Instituição, suas subsidiárias ou polos nela integrados / Teaching Staff integrated in Research Units of the Institution, its subsidiaries or integrated centers	7.6	100

5.4.6. Estabilidade e dinâmica de formação do corpo docente.

5.4.6. 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 de carreira com uma ligação à instituição por um período superior a três anos / Career teaching staff of the study programme with a link to the institution for over 3 years	6	78.947368421053	7.6
Docentes do ciclo de estudos inscritos em programas de doutoramento há mais de um ano (ETI) / FTE number of teaching staff registered in PhD programmes for over one year	0	0	7.6

Pergunta 5.5. e 5.6.

5.5. Procedimento de avaliação do desempenho do pessoal docente e medidas conducentes à sua permanente atualização e desenvolvimento profissional.

O procedimento de avaliação dos/as docentes da UC tem por base o disposto no “Regulamento de Avaliação de Desempenho dos Docentes da Universidade de Coimbra”. A avaliação do desempenho dos/as docentes da UC é efetuada relativamente a períodos de três anos e tem em consideração quatro vertentes: investigação; docência; transferência e valorização do conhecimento; gestão universitária e outras tarefas. O processo de avaliação compreende cinco fases (autoavaliação, validação, avaliação, audiência, homologação). O resultado final da avaliação de cada docente é expresso numa escala de quatro posições: excelente, muito bom, bom e não relevante. Antes de cada novo ciclo de avaliação, cada UO define, para as suas áreas disciplinares, o conjunto de parâmetros que determinam os novos objetivos do desempenho dos/as docentes e cada uma das suas vertentes, garantindo, assim, permanente atualização do processo.

5.5. Procedures for the assessment of the teaching staff performance and measures for their permanent updating and professional development.

The procedure for teaching staff performance assessment is stated in the “Regulamento de Avaliação de Desempenho dos Docentes da Universidade de Coimbra”. This assessment is performed every three years and covers four complementary dimensions: research, teaching, knowledge transfer and university management and other activities. The assessment process has five stages (self-assessment, validation, assessment, audience, and homologation). The final result of each individual assessment is expressed in a four position scale: excellent, very good, good and not-relevant. Before each new assessment cycle each Organic Unit defines, for its disciplinary areas, the set of parameters which determine the new performance objectives of their teaching staff and each one of the assessment dimensions, this ensuring a continuous update of the assessment process.

5.6. Observações: <sem resposta>

5.6. Observations: <no answer>

6. Pessoal Não Docente

6.1. Número e regime de tempo do pessoal não-docente afeto à lecionação do ciclo de estudos.

1x Secretariado do MSE [25% de afectação].

As atividades de natureza técnica e administrativa de apoio serão também asseguradas pelas/os funcionárias/os administrativos e atual corpo de técnicos da estrutura central da UC.

6.1. Number and work regime of the non-academic staff allocated to the study programme.

1x MSE secretary [25% allocation].

Support activities of a technical and administrative nature will also be ensured by the administrative staff and current staff of the UC's central structure.

6.2. Qualificação do pessoal não docente de apoio à lecionação do ciclo de estudos.

[1x] possuem licenciatura

6.2. Qualification of the non-academic staff supporting the study programme.

[1x] holds a BSc degree

6.3. Procedimento de avaliação do pessoal não-docente e medidas conducentes à sua permanente atualização e desenvolvimento profissional.

A Universidade de Coimbra garante uma avaliação do desempenho do seu pessoal não docente de acordo com o disposto na lei que rege o SIADAP que adotou o método de gestão por objetivos, estabelecendo uma avaliação do desempenho baseada na confrontação entre objetivos fixados e resultados obtidos. O processo de avaliação é bienal e concretiza-se: em reuniões com o/a avaliador/a, superior hierárquico/a imediato/a, para negociação e contratualização dos objetivos anuais e para comunicação dos resultados da avaliação; e no preenchimento de um formulário de avaliação. A avaliação visa identificar o potencial de desenvolvimento do pessoal e diagnosticar necessidades de formação. Para a aplicação do SIADAP, o processo é supervisionado pela Comissão Paritária e pelo Conselho Coordenador da Avaliação.

6.3. Assessment procedures of the non-academic staff and measures for its permanent updating and personal development

The University of Coimbra provides a assessment of its non-academic staff according to the laws related to SIADAP. This official regulation adopts an objective-based management method, establishing an assessment based on a comparison between goals established and results achieved. The assessment process is bi-annual and is established through meetings with the immediate hierarchic supervisor and the assessed individual to negotiate and establish the bi-annual goals as well as to inform the assessment result, and filling the assessment results form. This assessment aims at identifying the potential for personal development, and diagnose training needs; to apply SIADAP, the process is supervised by the Paritary Commission and the Assessment Coordination Council.

7. Instalações e equipamentos

7.1. Instalações físicas afetas e/ou utilizadas pelo ciclo de estudos (espaços letivos, bibliotecas, laboratórios, salas de computadores, etc.):

Sendo este um ciclo de estudos disponibilizado à distância, não lhe são especificamente afectas instalações físicas no campus. Contudo, como alunos da Universidade de Coimbra, os alunos deste ciclo de estudos poderão beneficiar e usufruir de todas as infraestruturas físicas disponíveis para todos os estudantes, nomeadamente espaços de estudo, reunião e videoconferência, bibliotecas, salas de computadores, estádio e ginásios universitários, Jardim Botânico, restaurantes universitários e cantinas.

7.1. Facilities used by the study programme (lecturing spaces, libraries, laboratories, computer rooms, ...):

Since this is a cycle of studies offered at a distance, no specific dedicated physical spaces are provided at the campus. However, as students of the University de Coimbra the MSE students can benefit from the common physical infrastructures available to all students such as study, meeting and videoconference spaces, libraries, computer rooms, university stadium and gyms, Botanic gardens, as well as university restaurants and canteens.

7.2. Principais equipamentos e materiais afetos e/ou utilizados pelo ciclo de estudos (equipamentos didáticos e científicos, materiais e TIC):

Este ciclo de estudos utilizará uma das várias salas de videoconferência disponíveis no Departamento de Engenharia Informática nomeadamente a sala A.5.1 para as actividades lectivas síncronas. O alojamento dos conteúdos far-se-á através dos servidores na nuvem do DEI e do Centro de Informática da Universidade; alguns

conteúdos, poderão ser disponibilizados através da plataforma educast da FCCN.

7.2. Main equipment or materials used by the study programme (didactic and scientific equipment, materials, and ICTs): *This study cycle shall use one of the many videoconference rooms available at the Department of Informatics Engineering, namely room A.5.1, for the synchronous lecturing activities. For hosting static online contents the DEI cloud or servers at the Computer Centre at the University of Coimbra shall be used; some online content might also be hosted at the FCCN educast platform.*

8. Atividades de investigação e desenvolvimento e/ou de formação avançada e desenvolvimento profissional de alto nível.

Pergunta 8.1. a 8.4.

8.1. Unidade(s) de investigação, no ramo de conhecimento ou especialidade do ciclo de estudos, em que os docentes desenvolvem a sua atividade científica.

<https://www.a3es.pt/si/iportal.php/cv/research-centers/formId/e36b1b42-eda4-2915-f50f-620f6e270c19>

8.2. Mapa-resumo de publicações científicas do corpo docente do ciclo de estudos, em revistas de circulação internacional com revisão por pares, livros ou capítulos de livro, relevantes para o ciclo de estudos, nos últimos 5 anos.

<https://www.a3es.pt/si/iportal.php/cv/scientific-publication/formId/e36b1b42-eda4-2915-f50f-620f6e270c19>

8.3. Mapa-resumo de atividades de desenvolvimento de natureza profissional de alto nível (atividades de desenvolvimento tecnológico, prestação de serviços ou formação avançada) ou estudos artísticos, relevantes para o ciclo de estudos:

<https://www.a3es.pt/si/iportal.php/cv/high-level-activities/formId/e36b1b42-eda4-2915-f50f-620f6e270c19>

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.

- *Promoting Sustainability as a Fundamental Driver in Software Development Training and Education, Erasmus+, 2020-09-01 to 2023-08-31*
- *BUBBLES - Defining the BUilding Basic BLocks for a U-Space SEparation Management Service, H2020-SESAR, 2020-06-01 to 2023-01-01*
- *VALU3S - Verification and Validation of Automated Systems' Safety and Security, H2020-ECSEL and FCT, 2020-05-01 to 2023-04-30*
- *ADVANCE - Addressing Verification and Validation Challenges in Future Cyber-Physical Systems, H2020, 2019-01-02 to 2022-12-30*
- *BASE - Biofeedback Augmented Software Engineering, FCT, 2018-07-26 to 2021-07-26*

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.

- *Promoting Sustainability as a Fundamental Driver in Software Development Training and Education, Erasmus+, 2020-09-01 to 2023-08-31*
- *BUBBLES - Defining the BUilding Basic BLocks for a U-Space SEparation Management Service, H2020-SESAR, 2020-06-01 to 2023-01-01*
- *VALU3S - Verification and Validation of Automated Systems' Safety and Security, H2020-ECSEL and FCT, 2020-05-01 to 2023-04-30*
- *ADVANCE - Addressing Verification and Validation Challenges in Future Cyber-Physical Systems, H2020, 2019-01-02 to 2022-12-30*
- *BASE - Biofeedback Augmented Software Engineering, FCT, 2018-07-26 to 2021-07-26*

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 o estudo da Direcção Geral de Estatísticas da Educação e Ciência (DGEEC) de Junho de 2020, utilizando dados do IEFP e do INE relativos aos desempregados registados no IEFP em junho de 2020 com habilitação superior obtida entre 2000 e 2020, e os diplomados entre 2000 e 2019 nos cursos que registam desemprego no IEFP.

Estes dados demonstram que nas áreas de engenharia de software, engenharia informática, e mesmo incluindo Sistemas e tecnologias não existe qualquer profissional registado como desempregado.

Estes dados, os indicadores europeus de carência de profissionais destas áreas (eurostat), os dados da Pordata, bem como a nosso conhecimento pessoal de extrema carência de profissionais nesta área são reveladores de um padrão robusto e incontornável de procura deste tipo de profissionais.

9.1. Evaluation of the employability of graduates by similar study programmes, based on official data:

According to the study by the Directorate-General for Education and Science Statistics (DGEEC) of June 2020, using data from the IEFP and the INE on the unemployed registered with the IEFP in June 2020 with a higher education between 2000 and 2020, and the graduates between 2000 and 2019 in courses that register unemployment in the IEFP.

These data still exist professionals in the fields of engineering software engineering, including systems and technologies that there are no registered professionals as engineering.

These data, the indicators of the shortage of professionals in these areas (seurostat), the data from Pordata, as well as our personal knowledge of the extreme need for robustness in this area, reveal a standard and unavoidable of this type of professionals.

9.2. Avaliação da capacidade de atrair estudantes baseada nos dados de acesso (DGES):

A procura elevada de profissionais com competências transversais em Engenharia de Software por parte do tecido empresarial, associada às boas condições de trabalho e oferta remuneratória competitiva, terá certamente um efeito sensível nas decisões dos estudantes à entrada do ensino superior. A experiência da FCTUC no lançamento de cursos ajustados às tendências do mercado e as competências nela existentes asseguram que tal oportunidade será concretizada com eficácia.

9.2. Evaluation of the capability to attract students based on access data (DGES):

The demand on professionals with transversal skills on Software Engineering and its continuous increase, together with the competitive average salary offered on in this field, will certainly have a sensible effect on students' decisions when selecting the course they will enroll. FCTUC's experience in launching courses in line with market trends and the skills therein, ensure that such an opportunity will be realized effectively.

9.3. Lista de eventuais parcerias com outras instituições da região que lecionam ciclos de estudos similares:

-

9.3. List of eventual partnerships with other institutions in the region teaching similar study programmes:

-

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:**

1) MSc Software Engineering, University of Hertfordshire Online, UK, 12 meses, 10.483€, <https://www.herts.ac.uk/courses/postgraduate-masters/msc-software-engineering-online/>

2) Master in Software Engineering, University of Limerick, Irlanda, 12 meses, 14.750€, <https://www.ul.ie/gps/course/software-engineering-msc>

3) MSc in Software Development, Queens University Belfast, Belfast UK, 1 ano, 25.871€, <https://www.qub.ac.uk/courses/postgraduate-taught/software-development-msc/>

4) Msc Software Engineering, University of Europe for Applied Sciences, Potsdam-Germany, 2, 3 ou 4 semestres | 13.000€/year, <https://web.ue-germany.com/msc-software-engineering-en-eu>

10.1. Examples of study programmes with similar duration and structure offered by reference institutions in the European Higher Education Area:

1) MSc Software Engineering, University of Hertfordshire Online, UK, 12 month, 10.483€, <https://www.herts.ac.uk/courses/postgraduate-masters/msc-software-engineering-online/>

2) Master in Software Engineering, University of Limerick, Irlanda, 12 month, 14.750€, <https://www.ul.ie/gps/course/software-engineering-msc>

3) MSc in Software Development, Queens University Belfast, Belfast UK, 1 year, 25.871€, <https://www.qub.ac.uk/courses/postgraduate-taught/software-development-msc/>

4) Msc Software Engineering, University of Europe for Applied Sciences, Potsdam-Germany, 2, 3 or 4 semestres | 13.000€/year, <https://web.ue-germany.com/msc-software-engineering-en-eu>

10.2. Comparação com objetivos de aprendizagem de ciclos de estudos análogos existentes em instituições de referência do Espaço Europeu de Ensino Superior:

A formação oferecida neste novo ciclo de estudos contém conteúdos programáticos idênticos às formações acima referidas. Existem contudo dois principais fatores diferenciadores:

1) a adoção da aprendizagem baseada em projecto e 2) a actualidade das unidades curriculares do MSE, tais como MSD (gestão de projeto de software assente em CMMI®), Methods (engenharia de requisitos) e Software Architectures, unidades curriculares que contém know-how mais atualizado fruto da cooperação com a Universidade de Carnegie-Mellon, sede do SEI (Software Engineering Institute), instituição de referência mundial na área da engenharia de Software.

10.2. Comparison with the intended learning outcomes of similar study programmes offered by reference institutions in the European Higher Education Area:

1) The adoption of project-based learning and 2) the state-of-art of the curricular units of MSE such as MSD (software project management based on CMMI®), Methods (requirements engineering) and Software Architectures, curricular units that include updated know as a result of the cooperation with Carnegie-Mellon's SEI (Software Engineering Institute), the worldwide reference in what concerns software 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 - N/A

11.1.1. Entidade onde os estudantes completam a sua formação:

N/A

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:

1. *Vasta experiência prévia do corpo docente no (actual) Mestrado em Engenharia de Software.*
2. *Forte ligação, alinhamento e apoio da indústria de software nacional.*
3. *Notoriedade já adquirida sobre a elevada qualidade desta oferta formativa.*
4. *Vasta experiência do corpo docente em formação remota baseada em projeto, incluindo avaliação individual.*
5. *Uma sólida infra-estrutura na UC e na FCCN para suportar a disponibilização de conteúdos remotos.*
6. *A maturidade do sistema desmaterializado de gestão académica e administrativa na UC, nomeadamente a verificação de identidade usando os mecanismos habituais utilizados pelos serviços desmaterializados da administração pública.*

12.1. Strengths:

1. *Vast previous experience of the teaching staff in the (current) Master in Software Engineering.*
2. *Close connection, alignment and support from relevant stakeholders of the national software industry.*
3. *Consolidated perception about the quality of the (current) master program.*
4. *Extensive experience of remote project-based education from the teaching staff, namely on individual student assessment.*
5. *The solid infrastructure of UC and FCCN for remote content delivery.*
6. *The consolidated on-line academic services of the University of Coimbra, namely its integration with the remote identity verification mechanisms provided by the Portuguese public administration.*

12.2. Pontos fracos:

Nada a assinalar

12.2. Weaknesses:

Nothing to report

12.3. Oportunidades:

1. *Manifesta e continuamente crescente necessidade da indústria global em profissionais com as competências oferecidas.*
2. *Disponibilidade e familiaridade do público-alvo (profissionais de software) em formação remota.*
3. *Beneficiar da 'vantagem do líder' em apresentar um MSc nacional intenso, de 12 meses.*
4. *Elevadas remunerações dos profissionais de IT, portanto com disponibilidade financeira para esta formação.*

12.3. Opportunities:

1. *The global software industry has a continuously increasing need of professionals with the competences this master program provides.*
2. *Readiness and familiarity of the target audience (software professionals) with remote learning.*
3. *Benefit of the first-mover advantage in providing a national intense 12-month program.*
4. *High wages of IT professionals, therefore with financial resources to apply for this master.*

12.4. Constrangimentos:

1. *Competição crescente por parte de grandes players nomeadamente Norte-Americanos.*
2. *Atractividade das remunerações da indústria de software que condicionam o campo de recrutamento de mentores externos.*
3. *Pouca visibilidade da Universidade de Coimbra fora do contexto nacional de indústria de software.*

12.4. Threats:

1. *Competition from big education&training players namely from the United States.*
2. *The high wages practiced at the software industry, which constrain our capability to attract external mentors.*
3. *Lack of visibility of the University of Coimbra outside the national software industry ecosystem.*

12.5. Conclusões:

Este novo ciclo de estudos surge como a evolução natural do actual Mestrado em Engenharia de Software (MSE) actualmente na sua segunda versão, para dar resposta aos constrangimentos identificados na sua actual formulação. Em concreto, os nossos alunos e vários potenciais candidatos têm referido sistematicamente dois problemas: 1) a exigência de presença física em algumas das unidades curriculares, o que está a ser mitigado com aulas em horário pós-laboral e 2) a longa duração da formação, dois anos, o que é incompatível com as dinâmicas desta área tecnológica. Devido à pandemia, neste momento temos vários alunos a estudar remotamente no Reino Unido, França, Suíça, e Áustria, bem como no Porto, Lisboa, Madeira e Viseu, com os problemas logísticos que as deslocações ou mudança para Coimbra lhes exigem. A experiência destes últimos dois anos lectivos preparou-nos

para assumir definitivamente o modelo de ensino à distância (o MSE original era oferecido remotamente a alunos nos Estados Unidos, Índia e Coreia do Sul). Na verdade, vemos esta reformulação como a única forma de ultrapassar os constrangimentos geográficos da Universidade de Coimbra, aumentando o campo de recrutamento de novos alunos e consequentemente fazer subir a qualidade global das candidaturas. Este ciclo de estudos continuará a contar com o apoio explícito da indústria de software nacional, em particular o cluster de IT criado na região Centro nos últimos anos, várias deles fundadas por alumni oriundos desta instituição e mesmo da primeira versão deste mestrado.

12.5. Conclusions:

This new cycle of studies is the natural evolution of the current Master in Software Engineering for working professionals (MSE) currently in its second version, tailored to address the constraints identified in its current model. Our students and numerous potential candidates have been consistently referring two main issues: 1) the mandatory requirement for in-person class attendance, which is being addressed by providing after-labour classes; 2) the duration of the master (2 years) which is incompatible with the dynamics of the IT technology sector. Due to the pandemic, we have currently students attending remotely from UK, France, Switzerland as well as from Lisbon, Porto, Madeira and Viseu, with the obvious logistic difficulties that travelling or relocation to Coimbra involves. The experience of these past two years has prepared us to assume decisively the distance-learning model (the original MSE version was delivered remotely to students in the USA, India and South Korea). In fact, we see this evolution as the only way to overcome the geographical constraints of the University of Coimbra, by increasing the reach for potential applications, thus increasing the average quality of individual candidates. This cycle of studies will continue to promote the support and direct involvement of the national software industry, namely from the IT cluster of the Center region of Portugal, several of which have been started by alumni of DEI from this same master program in its earlier editions.