

## Ficha de Unidade Curricular – (Versão A3ES 2018-2023)

### 1. Caracterização da Unidade Curricular.

- 1.1. **Designação da unidade curricular** (1.000 carateres).  
Laboratório de Software / Software Laboratory
- 1.2. **Sigla da área científica em que se insere** (100 carateres).  
IC
- 1.3. **Duração**<sup>1</sup> (100 carateres).  
Semestral
- 1.4. **Horas de trabalho**<sup>2</sup> (100 carateres).  
162 h
- 1.5. **Horas de contacto**<sup>3</sup> (100 carateres).  
Total - 67,5 h  
T - 18 h  
PL - 49,5 h
- 1.6. **ECTS** (100 carateres).  
6
- 1.7. **Observações**<sup>4</sup> (1.000 carateres).
- 1.7. **Remarks** (1.000 carateres).

2. **Docente responsável e respetiva carga letiva na Unidade Curricular** (preencher o nome completo) (1.000 carateres).  
Pedro Miguel Henriques dos Santos Félix

3. **Outros docentes e respetivas cargas letivas na unidade curricular** (1.000 carateres).

4. **Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes).** (1.000 carateres).

Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:

1. Desenvolver software aplicacional utilizando sistemas de gestão de bases de dados e interfaces Web, incluindo todos os elementos de apoio à sua manutenção e evolução, nomeadamente documentação técnica;
2. Usar ferramentas de suporte ao desenvolvimento de projetos de software, nomeadamente controlo de versões, automatização de tarefas de construção e testes unitários de software;
3. Usar metodologias de desenvolvimento ágeis, enquadradas no trabalho em equipas desenvolvimento de software em projetos de média dimensão;
4. Escrever um relatório técnico a descrever e enquadrar o trabalho realizado.

4. **Intended learning outcomes (knowledge, skills and competences to be developed by the students).** (1.000 characters).

Upon completing this unit, a student should be able to:

1. Develop software systems using relational database management systems and Web interfaces, including all the supporting elements for its maintenance and evolution, namely documentation.
2. Correctly use software development tools, such as version control systems, build automation tools and testing frameworks.
3. Use agile development methodologies, aligned with the work in software development teams of medium-sized projects.
4. Write a technical report describing and framing the work performed.

5. **Conteúdos programáticos (1.000 carateres).**

1. Sistemas de Controlo de Versões e formas de utilização (e.g. Git).
2. Organização do ambiente de desenvolvimento e processos de automatização no processo de construção de aplicações - "build automation" (e.g. Gradle).
3. Testes não supervisionados (unitários, funcionais e de integração) e plataformas de suporte (e.g. JUnit).
4. Utilização de aplicações de suporte ao desenvolvimento de projetos, nomeadamente nas componentes de controlo de versões, wiki e issue tracking (e.g. GitHub).

5. **Syllabus (1.000 characters).**

1. Version control systems and their associated usage practices (e.g. Git).
2. Development environment organization and build automation processes (e.g. Gradle).
3. Automatic software tests (unitary, functional and integration) and associated frameworks (e.g. JUnit).
4. Development supporting applications, namely wikis and issue trackers (e.g. GitHub).

6. **Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular (1.000 carateres).**

Esta unidade curricular tem como primeiro objetivo a consolidação dos conceitos e práticas fornecidos aos estudantes em unidades curriculares anteriores, nas área da programação e sistemas de informação, bem como melhorar a capacidade de escrita de relatórios técnicos.

Assim, o ponto 1. dos objetivos é assegurado através da utilização e aprofundamento de conhecimentos fornecidos nestas unidades curriculares anteriores.

O segundo propósito desta unidade curricular é dotar os estudantes das competências necessários ao trabalho em grupo (objetivo 3) e à utilização adequada de métodos e ferramentas utilizadas no desenvolvimento profissional de software (objetivo 2 e 3). Os conteúdos 1, 2, 3 e 4, bem como a sua utilização aplicada no contexto do desenvolvimento dum projeto, contribuem para estes objetivos. A escrita do relatório técnico sobre o trabalho desenvolvido contribuem para o objetivo 4.

6. **Evidence of the syllabus coherence with the curricular unit's intended learning outcomes (1.000 characters).**

The first goal of this curricular unit is the consolidation of concepts and practices provided in previous curricular units, namely in the programming and information systems areas. So, this first learning outcome is ensured by the usage and refinement of the theoretical and practical subjects provided in these previous units. The second goal is to provide the students with the ability to work in development teams (goal 2.) and correctly use the concepts and practices associated with professional software development (goals 2 and 3.). The syllabus contents 1, 2, 3 and 4, as well as their usage in an applied context, are the main contributors for this goal. The writing of the technical report on the work carried out contributes to objective 4.

7. **Metodologias de ensino (avaliação incluída) (1.000 carateres).**

Ensino realizado em ambiente de prática laboratorial (30 aulas, 67,5 horas de contacto). O tempo total de trabalho do estudante é de 162 horas. As primeiras 2 semanas são usadas para a apresentação teórica dos conteúdos programáticos. Uma semana intermédia é usada para a avaliação individual e apresentação do trabalho realizado até ao momento. As restantes 12 semanas são usadas no desenvolvimento iterativo de um projeto de software, dividido em quatro etapas de 3 semanas cada e realizado em grupos de 2 a 3 alunos. Durante estas, as horas de contacto são usadas para orientar os alunos no desenho, implementação e organização do projeto, bem como para apresentar ou complementar temas necessários à sua realização. Esta atividade é realizada em ambiente laboratorial e é também usada para a avaliação contínua dos alunos. No final, a avaliação final do projeto é realizada individualmente com cada grupo de trabalho, tendo em conta todos os elementos produzidos, incluindo a documentação. A avaliação final de cada aluno é baseada na classificação do projeto realizado ao longo das 12 semanas e na contribuição individual do aluno no projeto.

7. **Teaching methodologies (including assessment) (1.000 characters).**

This unit is taught in a laboratory environment (30 lectures, 67.5 contact hours). The students are expected to use a total of 162 hours for the successful completion of this unit.

The first 2 weeks are used for the theoretical presentation of the syllabus contents.

An intermediate week is used for individual student evaluations as well as for the students to showcase the

work done until that moment.

The remaining 12 weeks are used in the iterative development of a software project, divided into 4 stages with 3 weeks each. This project is developed in groups of 2 to 3 students. During these 12 weeks, the contacts hours are used primarily to supervise the students on the design, implementation and organization of their project; as well as to present or complement additional topics required for the projects. This activity is done in a laboratory environment and is also used to the continuous student evaluation.

At the end, there is an interactive final project evaluation, performed individually for each student group.

The final student grade is based on the project grade and on the individual contribution to the project.

**8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular (3.000 carateres).**

O ensino desta unidade curricular é maioritariamente realizado em ambiente laboratorial, consistindo no esclarecimento e aconselhamento dos alunos sobre os desafios do projeto e da sua organização. A existência de quatro etapas, com entrega de versões funcionais do projeto por parte dos grupos de trabalho, proporciona a existência de vários momentos de avaliação. Nestes, os docentes fazem uma análise crítica do trabalho já realizado e perspetivam melhoramentos.

Considera-se que este ensino e avaliação iterativa, baseada em trabalho realizado pelos alunos, é adequado aos objetivos da unidade curricular, nomeadamente os pontos 1 e 2. A realização em grupo de um projeto com a duração de 12 semanas e várias versões é adequada ao objetivo 3. A documentação produzida é adequada ao objectivo 4.

**8. Evidence of the teaching methodologies coherence with the curricular unit's intended learning outcomes (3.000 characters).**

This unit is taught mainly in a laboratory environment, by supervising and guiding the students through the software development challenges. The division of the project into 4 stages, with incremental delivery of working versions, allows for multiple evaluation phases. In those, the teaching staff provides feedback on the previously delivered versions and ideas for future improvements.

This incremental evaluation and guidance process, based on concrete work developed by the students, provides for the unit's learning outcomes, namely items 1 and 2. The development of a group project with twelve-week length and multiple functional versions provides for item 3 of the learning outcomes. The documentation produced is suitable for objective 4.

**9. Bibliografia de consulta/existência obrigatória (1.000 carateres).**

Conjunto de notas disponibilizadas em repositório público (e.g. <https://github.com/isel-leic-ls/1819-1-common/wiki>), incluindo referências para conteúdos externos, também de acesso público (e.g. S. Chacon, B. Straub, "Pro Git", Apress, 2018).

Bibliografia das unidades curriculares anteriores, nas áreas da programação e dos sistemas de informação.

---

<sup>1</sup> Anual, semestral, trimestral, ...

<sup>2</sup> Número total de horas de trabalho.

<sup>3</sup> Discriminadas por tipo de metodologia adotado (T - Ensino teórico; TP - Ensino teórico-prático; PL - Ensino prático e laboratorial; TC - Trabalho de campo; S - Seminário; E - Estágio; OT - Orientação tutorial; O - Outro).

<sup>4</sup> Assinalar sempre que a unidade curricular seja optativa.