

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

1. Caracterização da Unidade Curricular.

1.1. Designação da unidade curricular (1.000 carateres).

Técnicas de Desenvolvimento de Software / Software Development Techniques

1.2. Sigla da área científica em que se insere (100 carateres).

IC

1.3. Duração¹ (100 carateres).

Semestral

1.4. Horas de trabalho² (100 carateres).

162 h

1.5. Horas de contacto³ (100 carateres).

Total - 67,5 h

T - 43,5 h

TP - 12 h

PL - 12 h

1.6. ECTS (100 carateres).

6

1.7. Observações⁴ (1.000 carateres).

Comum com outros cursos

1.7. Remarks (1.000 carateres).

Common with other courses

2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo) (1.000 carateres).

Pedro Alexandre de Seia e Cunha Ribeiro Pereira

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. Demonstrar conhecimento sobre os princípios aplicáveis no âmbito do desenvolvimento de software;
2. Identificar as propriedades pretendidas no software e demonstrar conhecimento acerca das técnicas a usar para as influenciar;
3. Analisar e justificar a aplicação de padrões de desenho em contextos selecionados;
4. Desenhar, implementar e testar aplicações interativas de média dimensão;
5. Produzir documentação técnica onde se justificam as decisões tomadas no software desenvolvido.

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

Students who successfully complete this course unit should be able to:

1. Demonstrate know-how on software development principles;
2. Identify software properties and demonstrate know how on the techniques used to influence them;
3. Analyze and justify the application of design patterns in selected contexts;
4. Design, implement and test medium sized interactive applications;
5. Write technical documentation justifying decisions made in the context of software development.

5. Conteúdos programáticos (1.000 carateres).

- I. Princípios elementares de *type driven development*: encapsulamento e especificação de contratos e invariantes.

- II. Princípios elementares de *test driven development*: especificação explícita das dependências e respetiva resolução através de injeção.
- III. Composição de funções: *high order functions*, *currying* e aplicação parcial.
- IV. Composição de tipos: agregação, herança e polimorfismo.
- V. Princípios gerais de desenho de software: DRY, *Law of Demeter* e SOLID.
- VI. Seleção de padrões de desenho de uso geral.
- VII. Seleção de padrões de desenho usados no âmbito do desenvolvimento de aplicações interativas.

5. Syllabus (1.000 characters).

- I. Basic principles of type driven development: encapsulation and contract specification and its invariants;
- II. Basic principles of test-driven development: explicit specification of dependencies and their resolution through dependency injection;
- III. Function composition: high order functions: currying and partial application;
- IV. Type composition: aggregation, inheritance and polymorphism;
- V. General principles of software development: DRY, Law of Demeter and SOLID;
- VI. Selection of design patterns of general usage;
- VII. Selection of design patterns used in the context of interactive applications.

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

Esta unidade curricular introduz os conceitos e o vocabulário fundamental dos paradigmas e dos princípios usados no desenvolvimento de software.

Os objetivos 1, 2 e 3 são alcançados através dos itens I até VII do conteúdo programático.

O desenvolvimento de programas durante as sessões práticas permite atingir os objetivos 4 e 5.

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

This curricular unit introduces the fundamental concepts and the vocabulary of the paradigms and principles used in software development.

Goals 1, 2 and 3 are achieved through items I to VII of the syllabus.

The development of programs during practical sessions makes it possible to achieve goals 4 and 5 of learning outcomes.

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

Previstas 30 aulas durante o semestre (15 de 3 horas e 15 de 1,5 horas). As aulas destinam-se à apresentação dos temas e de exemplos práticos de aplicação. Os tópicos principais são ainda explorados através da realização de trabalhos práticos para desenvolver programas interativos.

Os resultados da aprendizagem 1, 2 e 3 são avaliados individualmente através do teste final escrito e das fichas realizadas durante o semestre. Durante o acompanhamento dos trabalhos de grupo realizados nas aulas práticas são avaliados os resultados da aprendizagem 4 e 5. Os resultados da aprendizagem 4 e 5 também são avaliados na discussão final dos trabalhos de grupo.

Nota final corresponde a 40% do teste final, 40% dos trabalhos e 20% das fichas realizadas nas aulas.

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

Planned during the semester in 30 lectures (15 of 3 hours and 15 of 1.5 hours). The lectures are intended for presentation of topics and practical examples. The main topics are further explored through practical work to develop Java programs using graphical user interface.

Learning outcomes 1, 2 and 3 are individually assessed through short written tests during the semester and final test. During the monitoring of group work performed in class practices are assessed learning outcomes 4 and 5. Learning outcomes 4 and 5 are also evaluated in the final discussion of work group.

Final grade corresponds to 40% of the final test, 40% of the practical works and 20% of the evaluation sheets made in class.

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

O conhecimento das técnicas de programação event-driven e dos princípios usados no desenvolvimento de software é obtido através de aulas interativas e respetivos elementos de apoio, e da realização de uma aplicação, com a participação dos alunos, que necessita de especializar comportamentos, reutilizar código e

utilizar interface com o utilizador, textual ou gráfica.

A competência para desenvolver boas práticas de desenho e de escrita de código para aplicações é desenvolvida através da realização de trabalhos com supervisão e da sua avaliação crítica.

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

Knowledge of event-driven programming techniques and the principles used in software development is obtained through interactive classes and respective support elements, and through an application, with the participation of students, that needs to specialize behaviors, reuse code and use a user interface, textual or graphical.

The competence to develop good design and code writing practices for applications is developed through supervised work and critical evaluation.

9. Bibliografia de consulta/existência obrigatória (1.000 caracteres).

A. Soshin, Hands-on Design Patterns with Kotlin, Packt publishing, 2018. ISBN-13: 978-1788998017; ISBN-10: 1788998014

¹ Anual, semestral, trimestral, ...

² Número total de horas de trabalho.

³ 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).

⁴ Assinalar sempre que a unidade curricular seja optativa.