
1. Designação da unidade curricular

[2179] Engenharia de Software / Software Engineering

2. Sigla da área científica em que se insere

IC

3. Duração

Unidade Curricular Semestral

4. Horas de trabalho

162h 00m

5. Horas de contacto

Total: 60h 00m das quais T: 30h 00m | TP: 10h 00m | P: 20h 00m

6. % Horas de contacto a distância

Sem horas de contacto à distância

7. ECTS

6

8. Docente responsável e respetiva carga letiva na Unidade Curricular

[838] Luís Filipe Graça Morgado | Horas Previstas: 60 horas

9. Outros docentes e respetivas cargas letivas na unidade curricular

[2128] Jorge Boto Viegas Branco | Horas Previstas: 60 horas

10. Objetivos de aprendizagem e a sua compatibilidade com o método de ensino (conhecimentos, aptidões e competências a desenvolver pelos estudantes).

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

1. Compreender e descrever os princípios base de desenvolvimento de sistemas, com aplicação específica aos processos de desenvolvimento de software, bem como os métodos de modelação, projecto, implementação e verificação que suportam esses processos de desenvolvimento;
2. Identificar e relacionar as principais características de diferentes processos de desenvolvimento de software em termos de domínios de aplicação, potencialidades e limitações;
3. Utilizar processos e métodos de desenvolvimento sistemáticos, disciplinados e quantificáveis na concepção e desenvolvimento de software;
4. Verificar e testar os sistemas desenvolvidos tendo em conta garantir a qualidade dos produtos finais.

10. Intended Learning objectives and their compatibility with the teaching method (knowledge, skills and competences by the students).

Students who successfully complete this course will be able to:

1. Understand and describe the basic principles of systems development, with specific application to software development processes, and the methods of software modeling, design, implementation and verification that support such development processes;
2. Identify and relate the main features of various software development processes in terms of application domains, capabilities and limitations;
3. Use systematic, disciplined and quantifiable processes and methods in the design and development of software;
4. Verify and test the developed systems ensuring the quality of the final products.

11. Conteúdos programáticos

- I. Introdução à análise e desenvolvimento de sistemas.
- II. Processos de desenvolvimento de software.
- III. Análise e especificação de requisitos.
- IV. Modelação e arquitetura de software.
- V. Implementação, teste e qualidade.

11. Syllabus

- I. Introduction to systems analysis and development.
- II. Software development processes.
- III. Requirements analysis and specification.
- IV. Software modelling and architecture.
- V. Implementation, test and quality.

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

Esta disciplina visa o estudo de abordagens sistemáticas, disciplinadas e quantificáveis de desenvolvimento de software (objectivos 1 e 2 concretizados em I e II) desenvolvendo nos alunos a capacidade de realizar a análise de problemas concretos e a capacidade de modelação, projecto, implementação e teste dos sistemas respectivos, tendo por base processos de desenvolvimento sistemáticos (objectivos 3 e 4 concretizados em III, IV e V).

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

This course aims to study systematic, disciplined and quantifiable approaches to software development (objectives 1 and 2 achieved in I and II) developing in the students the ability to perform the analysis of concrete problems and the ability of modeling, design, implementation and testing of the resulting systems, based on systematic development processes (objectives 3 and 4 realized in III, IV and V).

13. Metodologias de ensino e de aprendizagem específicas da unidade curricular articuladas com o modelo pedagógico

É utilizada uma metodologia de ensino teórico-prática, suportada num projecto desenvolvido ao longo do semestre.

13. Teaching and learning methodologies specific to the curricular unit articulated with the pedagogical model

A theoretical-practical teaching methodology is used, supported by a project developed throughout the semester.

14. Avaliação

A avaliação é distribuída sem exame final.
Os objectivos de aprendizagem são avaliados com base na realização e entrega obrigatória de um projecto individual (60%) e na realização e entrega obrigatória de um relatório individual (40%), cuja classificação é obtida pelo conhecimento demonstrado nas entregas realizadas e por uma discussão do projecto e do relatório.
Todas as componentes são pedagogicamente fundamentais.

14. Assessment

Assessment is distributed without a final exam.
The learning objectives are assessed based on the completion and mandatory delivery of an individual project (60%) and the completion and mandatory delivery of an individual report (40%), the classification of which is obtained by the knowledge demonstrated in the deliveries made and by a discussion of the project and the report.
All components are pedagogically fundamental.

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

Os temas correspondentes aos objectivos de aprendizagem, na sua componente conceptual, são estudados em aulas teóricas específicas, sendo concretizados em casos práticos e no projecto desenvolvido ao longo do semestre em aulas teórico-práticas e de modo autónomo pelos alunos.

15. Evidence of the teaching methodologies coherence with the curricular unit's intended learning outcomes

The themes related to the learning outcomes, in their conceptual component, are studied in specific lectures, and concretized by practical problems and by the project developed during the semester in practical classes and independently by students.

16. Bibliografia de consulta/existência obrigatória

1-R. Pressman, B. Maxim (2020), Software Engineering: A Practitioner's Approach, McGraw Hill
2-C. Larman (2004), Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development, Prentice Hall

17. Observações

Unidade Curricular Obrigatória
Unidade Curricular comum ao(s) curso(s) de MEIM

Data de aprovação em CTC: Data de aprovação em CTC:

Data de aprovação em CP: Data de aprovação em CP: