
Ano Letivo 2020-21

Unidade Curricular ENGENHARIA DE SOFTWARE

Cursos ENGENHARIA INFORMÁTICA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14781052

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Línguas de Aprendizagem Português

Modalidade de ensino Presencial

Docente Responsável Marielba Silva de Zacarias

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Marielba Silva de Zacarias	PL; T	T1; PL1; PL2; PL3	28T; 84PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
2º	S2	28T; 28PL	156	6

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Conhecimentos sobre o paradigma orientado por objetos

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Estudar o processo completo de desenvolvimento de um sistema, desde a sua fase de conceção até à sua entrega e manutenção. Para cada uma destas fases, são identificados os principais problemas e as técnicas conhecidas para os abordar. No final da unidade curricular, os alunos devem:

- Conhecer e justificar as várias fases de desenvolvimento de um sistema de informação.
- Identificar problemas associados a cada fase e as técnicas conhecidas para os abordar.
- Conhecer e aplicar formalismos para a análise de requisitos
- Planear e gerir o desenvolvimento de um sistema de informação de média dimensão

Conteúdos programáticos

Introdução à Engenharia de Software

- Problemática do desenvolvimento de software
- Paradigmas da Engenharia de Software
- Fases do processo de desenvolvimento de software

Modelos de Processos de Desenvolvimento de Software

- Conceitos gerais sobre Modelos de Processos de Desenvolvimento
- Modelos tradicionais de Processos de Desenvolvimento de Software
- Processos Ágeis

Engenharia de Requisitos

- Processo de Engenharia de Requisitos
- Levantamento de Requisitos
- Técnicas de modelação: Casos de Utilização
- Técnicas de Prototipagem
- Documentação de Requisitos

Gestão de Projetos tradicionais e ágeis

- Conceitos de Gestão de Projeto
- Métricas de Processo e Projeto
- Estimativa de Projetos de Software
- Planeamento de Projetos de Software
- Gestão de Risco
- Gestão de Qualidade
- Gestão de Configuração

Metodologias de ensino (avaliação incluída)

Aulas Teóricas: As noções teóricas serão dadas por método predominantemente expositivo, com projeção e explicação dos objetivos e conteúdos correspondentes a cada tema, acompanhado de debate, colocação e esclarecimento de dúvidas.

Aulas Práticas: Os estudantes serão motivados a aplicar as competências adquiridas através de atividades práticas, incluindo a análise e implementação de problemas.

A avaliação considera provas escritas (PE) e trabalho(s) prático(s) (TP). As provas escritas PE são os exames de época normal, de recurso, época para estudantes com estatuto especial ou época especial de conclusão de curso e é exigida a nota mínima de 6 valores.

A admissão a exame:

1. requer assiduidade a 75% das aulas teóricas e 80% das aulas práticas;
2. tem por condição a entrega obrigatória das várias componentes do trabalho prático, com nota mínima de 8 em cada componente;
3. requer avaliação final TP $\geq 9,5$.

Se PE < 6 então notal final = PE

Caso contrário a nota final = 60% PE + 40% TP

Bibliografia principal

- Software Engineering: A Practitioner's Approach, 8th Edition, 2014, Roger S Pressman, McGraw-Hill Higher Education
- Requirements Engineering - Fundamentals, 1st Edition, 2010, Principles, and Techniques, Klaus Pohl, Springer-Verlag Berlin Heidelberg
- Software Engineering: Theory and Practice, 4th Edition, 2009, Shari Lawrence Pfleeger, Joanne M. Atlee, Prentice Hall
- UML-Metodologias e Ferramentas CASE, volume 1, 2a Edição, 2005, Alberto Silva e Carlos Videira, Centro Atlântico
- Introdução à Engenharia de Software, 1a Edição, 2015, Sérgio Guerreiro, FCA-Editora de Informática

Academic Year 2020-21

Course unit SOFTWARE ENGINEERING

Courses INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area

Acronym

Language of instruction Portuguese

Teaching/Learning modality In classroom.

Coordinating teacher Marielba Silva de Zacarias

Teaching staff	Type	Classes	Hours (*)
Marielba Silva de Zacarias	PL; T	T1; PL1; PL2; PL3	28T; 84PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
28	0	28	0	0	0	0	0	156

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

Knowledge of the object-oriented paradigm

The students intended learning outcomes (knowledge, skills and competences)

This curricular unit aims the study of the process of developing software systems, from conceptual phase through to delivery and maintenance. For each of these phases, main problems are identified as well as known techniques to address them. At the end of the course, students should:

- Know and be able to justify the several and different phases of an entire software development process.
- Be able to identify problems associated with each software development phase as well as the techniques to address them.
- Know and be able to apply requirements analysis formalisms.
- Be able to plan and manage middle size software development projects.

Syllabus

Introduction to Software Engineering

- Software Development Problem
- Software Engineering Paradigms
- Phases of the Software Development Process

Software Development Process Models

- General Concepts about Software Development Process Models
- Traditional Software Development Process
- Agile Processes

Requirements Engineering

- Requirements Engineering Process
- Requirements Elicitation and Analysis
- Modelling Techniques: Use Case
- Prototyping Techniques
- Requirements Documentation

Traditional and Agile Project Management

- Project Management Concepts
- Process and Project Metrics
- Estimation for Software Projects
- Software Project Scheduling
- Risk Management
- Software Quality Management
- Software Configuration Management

Teaching methodologies (including evaluation)

Theoretical lessons: Theoretical notions are predominantly addressed by expository-style lectures, projection and explanation of objectives and contents relevant to each theme, followed by debate and questions.

Practical lessons: Students are encouraged to apply the competences acquired through practical activities, including the analysis and development of problems.

The assessment considers written exams (WE) and practical work(s) (PW). A minimum grade of 6 is required in the written exams (all seasons).

Admission to the exam:

- requires attendance to 75% of the theoretical classes and 80% of the practical classes;
- subject to the mandatory delivery of the PW's several components with the minimum grade of 8 in each component;
- requires final PW ≥ 9.5 .

If $WE < 6$ then final grade = WE

Otherwise the final grade = 60% WE + 40% PW

Main Bibliography

- Software Engineering: A Practitioner's Approach, 8th Edition, 2014, Roger S Pressman, McGraw-Hill Higher Education
- Requirements Engineering - Fundamentals, 1st Edition, 2010, Principles, and Techniques, Klaus Pohl, Springer-Verlag Berlin Heidelberg
- Software Engineering: Theory and Practice, 4th Edition, 2009, Shari Lawrence Pfleeger, Joanne M. Atlee, Prentice Hall
- UML-Metodologias e Ferramentas CASE, volume 1, 2a Edição, 2005, Alberto Silva e Carlos Videira, Centro Atlântico
- Introdução à Engenharia de Software, 1a Edição, 2015, Sérgio Guerreiro, FCA-Editora de Informática