
Ano Letivo 2020-21

Unidade Curricular LABORATÓRIO DE ENGENHARIA DE SOFTWARE

Cursos ENGENHARIA INFORMÁTICA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14781064

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Línguas de Aprendizagem Português

Modalidade de ensino Presencial ou videoconferência (plano de contingência COVID-19)

Docente Responsável Paula Cristina Negrão Ventura Martins

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Paula Cristina Negrão Ventura Martins	PL; T	T1; PL1; PL2	14T; 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
3º	S2	14T; 42PL	156	6

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

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Programação, Engenharia de Software, Análise e Modelação de Sistemas, Interfaces Pessoa-Máquina

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Aprofundar a cultura e sensibilidade acerca das temáticas da engenharia dos sistemas de informação, designadamente nos aspectos avançados da modelação, do processo de desenvolvimento e da utilização de ferramentas CASE de suporte. No final, os alunos devem:

- Descrever e aplicar as várias fases do processo de desenvolvimento de software
- Planear e gerir o desenvolvimento ágil de um sistema informático
- Conhecer e aplicar ferramentas no desenvolvimento de aplicações de acordo com os padrões da indústria
- Desenvolver e testar o sistema informático de acordo com as boas práticas do processo selecionado e da engenharia de software

Conteúdos programáticos

1. Metodologias de desenvolvimento de software tradicionais (Iconix, RUP, Catalysis, Nikel, Ferramentas EPF)
2. Metodologias de desenvolvimento de software ágeis (XP, Scrum, Crystal, DSDM)
3. Metamodelo SPEM
4. Melhoria do Processo de Desenvolvimento de Software (CMMI, ISO/IEC TR 15504 SPICE, BOOTSTRAP)
5. Ferramentas Colaborativas de Gestão de Projectos
6. Ferramentas CASE (Evolução histórica, áreas de intervenção, classificações, avaliação)
7. Model-Driven Approach (MDA) e Model-Driven Development (MDD)
8. Testes de software
9. Gestão de Processos de Negócio

Metodologias de ensino (avaliação incluída)

Unidade curricular sem exame. Avaliação por trabalho teórico e projeto final.

Aulas T(14h) 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 P(42h): Os estudantes serão motivados para aplicar as competências adquiridas através de atividades práticas, incluindo a análise e implementação de problemas.

Desenvolvimento de sistema informático de média dimensão integrando os conhecimentos disciplinas da área científica de Sistemas de Informação e Bases de Dados e recorrendo a metodologias ágeis

Reuniões semanais para gestão do projeto

A aprovação depende da assiduidade: (75% aulas teóricas) e (90% aulas laboratoriais). Componentes de avaliação: Apresentação (15%), Documentação (15%), Gestão Projecto (10%), Implementação e testes (60%). Classificação entre 0-20 valores.

Bibliografia principal

1. Software Engineering: A Practitioner's Approach, 8th Edition, 2014, Roger S Pressman, MacGraw-Hill Higher Education
2. Software Engineering: Theory and Practice, 4th Edition, 2019
3. Shari Lawrence Pfleeger, Joanne M. Atlee, Prentice Hall
4. CMMI(R): Guidelines for Process Integration and Product Improvement, 3rd Edition, 2011
5. Mary Beth Chrissis, Mike Konrad, Sandy Shrum, Addison-Wesley
6. Model Driven Architecture (OMG): Applying MDA to Enterprise Computing, David S. Frankel, Wiley, 2010
7. UML-Metodologias e Ferramentas CASE, Alberto Silva e Carlos Videira, 2ª Edição, 2005, Centro Atlântico

Academic Year 2020-21

Course unit SOFTWARE ENGINEERING LABORATORY

Courses INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area

Acronym

Language of instruction Portuguese

Teaching/Learning modality Presential or videoconference (COVID-19 contingency plan)

Coordinating teacher Paula Cristina Negrão Ventura Martins

Teaching staff	Type	Classes	Hours (*)
Paula Cristina Negrão Ventura Martins	PL; T	T1; PL1; PL2	14T; 84PL

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

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
14	0	42	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

Programming, Software Engineering, Systems Analysis and Modeling, Human-Computer Interaction

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

This curricular unit aims to deepen and explore the culture and sensibility about software engineering issues, particularly advanced aspects of modelling, software development processes and CASE tools. In the end, students should be able to:

1. Describe and apply the several software development phases
2. Plan and manage a software development project
3. Know and use appropriate software development tools and management tools to develop applications conforming to industry standards
4. Develop and test a software system according to the software development process best practices.

Syllabus

1. Traditional Software Development Methodologies (Iconix, RUP, Catalysis, Nikel, Ferramentas EPF)
2. Agile Software Development Methodologies (XP, Scrum, Crystal, DSDM)
3. SPEM Meta-model
4. Software Development Process Improvement (CMMI, ISO/IEC TR 15504 SPICE, BOOTSTRAP)
5. Collaborative Project Management Tools
6. CASE Tools (Evolution, intervention areas, classification, evaluation)
7. Model-Driven Approach (MDA) e Model-Driven Development (MDD)
8. Software Tests
9. Business Process Management

Teaching methodologies (including evaluation)

Curricular unit without an exam. Evaluation by theoretical work and a final project.

Theoretical lessons (14h). Theoretical notions are predominantly given by expository-style lectures, projection and explanation objectives and contents relevant to each theme, followed by debate and questions.

Practical lessons (42h). Students are encouraged to apply the competences acquired through practical activities, including the analysis and development of problems. Development of a medium-sized software system, in conjunction with skills and concepts from previous courses in the scientific area of Information Systems and Databases using agile methodologies. Weekly meetings for project management.

Course approval depends on the following attendance conditions: 75% of theoretical lessons and 90% of lab lessons. Assessment components: Presentation (15%), Documentation (15%), Project Management (10%), Implementation and tests (60%). Classification from 0-20 values.

Main Bibliography

1. Software Engineering: A Practitioner's Approach, 8th Edition, 2014, Roger S Pressman, MacGraw-Hill Higher Education
2. Software Engineering: Theory and Practice, 4th Edition, 2019
3. Shari Lawrence Pfleeger, Joanne M. Atlee, Prentice Hall
4. CMMI(R): Guidelines for Process Integration and Product Improvement, 3rd Edition, 2011
5. Mary Beth Chrissis, Mike Konrad, Sandy Shrum, Addison-Wesley
6. Model Driven Architecture (OMG): Applying MDA to Enterprise Computing, David S. Frankel, Wiley, 2010
7. UML-Metodologias e Ferramentas CASE, Alberto Silva e Carlos Videira, 2ª Edição, 2005, Centro Atlântico