
Ano Letivo 2022-23

Unidade Curricular TÓPICOS AVANÇADOS EM ENGENHARIA DE SOFTWARE

Cursos ENGENHARIA INFORMÁTICA (2.º ciclo)

CIBERSEGURANÇA (*)

(*) Curso onde a unidade curricular é opcional

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14741036

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Código CNAEF (3 dígitos) 523

**Contributo para os Objetivos de
Desenvolvimento Sustentável - 9;12;17
ODS (Indicar até 3 objetivos)**

Línguas de Aprendizagem

Inglês

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	28T; 28PL

* 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
1º	S1	28T; 28PL	156	6

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

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Programação orientada por objectos; Análise e Modelação de Sistemas (UML); Engenharia de Software

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Esta unidade curricular tem por objectivo proporcionar contacto com os paradigmas mais recentes no que se refere ao desenvolvimento de software (MDSD), sendo uma abordagem direccionada para a aplicação de modelos. O Desenvolvimento conduzido por modelos recorre a linguagens de domínio específico (DSL) para criar os modelos que descrevem a estrutura e comportamento das aplicações. No final, os mestrandos deverão estar aptos para:

1. Aplicar técnicas de transformação de modelos;
 2. Conhecer o processo de Engenharia para Desenvolvimento Baseado em Modelos;
 3. Conhecer e usar ferramentas de MDSD;
 4. Dado um problema num domínio específico, desenvolver uma DSL
-

Conteúdos programáticos

1. Os objectivos do desenvolvimento baseado em modelos
2. Terminologia
3. Abordagens baseadas em Modelos (Model-Driven Architecture)
4. Meta-modelação
 - Conceitos
 - Arquitetura de 4 camadas OMG
 - Meta-Meta-Modelação
 - MOF e UML
 - Perfis UML
 - OCL
 - XMI
5. Fábricas de Software
6. Processos de Engenharia
7. Transformação de modelos
8. Transformação modelo-para-texto
9. Transformação modelo-para-modelo
10. Linguagens de Domínio Específico
11. Ferramentas MDD (Model-Driven Development)

Metodologias de ensino (avaliação incluída)

Aulas Teóricas : As noções teóricas serão dadas por método predominantemente expositivo, com projeção e explicação dos objectivos 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:

- tem por condição a entrega obrigatória das várias componentes do TP, com nota mínima de 8 em cada componente;
- requer avaliação final TP $\geq 9,5$.

Se PE < 6 então nota final = PE

Caso contrário a nota final = 50% PE + 50% TP

Bibliografia principal

1. Model-Driven Software Engineering in Practice, Marco Brambilla, Jordi Cabot, Manuel Wimmer, Morgan & Claypool Publishers, 2012
2. Model-Driven Software Development: Technology, Engineering, Management, Thomas Stahl, Markus Voelter, Wiley, 2006
3. Software Language Engineering: Creating Domain-Specific Languages Using Metamodels, Anneke Kleppe, Addison Wesley, 2008
4. Domain-Specific Modeling, Steven Kelly, Juha-Pekka Tolvanen, Wiley-Blackwell, 2008

Academic Year 2022-23

Course unit ADVANCED TOPICS IN SOFTWARE ENGINEERING

Courses INFORMATICS ENGINEERING

Common Branch
(*)

(*) Optional course unit for this course

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area

Acronym

CNAEF code (3 digits) 523

**Contribution to Sustainable
Development Goals - SGD
(Designate up to 3 objectives)** 9;12;17

Language of instruction English

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	28T; 28PL

* 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

Object-Oriented Programming; System Analysis and Design (UML); Software Engineering.

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

This curricular unit aims to provide contact with highly regarded development paradigms concerning the usage of models. Model-Driven Software Development (MDSO) aims to use domain-specific languages (DSL) to create models that express application structure and behaviour. In the end, master students should be able to:

1. Apply techniques of model transformation;
2. Know Model-Driven Engineering Methodologies;
3. Know and use MDSO tools;
4. Given a specific domain problem, develop a DSL.

Syllabus

1. The rationale for Model-Driven Software Development
2. Terminology
3. Model-Driven Architecture (MDA)
4. Meta-Modelling
 - Concepts
 - OMG 4 layers of architecture
 - Meta-Meta-Modelling
 - UML profiling
 - Object Constraint Language (OCL)
 - Meta-data interchange and serialization (XMI)
5. Software Factories
6. Model-driven engineering methodologies
7. Model Transformations
8. Model-to-Text Transformations
9. Model-to-Model Transformations
10. Domain-Specific Languages
11. MDSD Tools

Teaching methodologies (including evaluation)

Theoretical lessons: Theoretical notions are predominantly given 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:

- 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 = 50% WE + 50% PW

Main Bibliography

1. Model-Driven Software Engineering in Practice, Marco Brambilla, Jordi Cabot, Manuel Wimmer, Morgan & Claypool Publishers, 2012
2. Model-Driven Software Development: Technology, Engineering, Management, Thomas Stahl, Markus Voelter, Wiley, 2006
3. Software Language Engineering: Creating Domain-Specific Languages Using Metamodels, Anneke Kleppe, Addison Wesley, 2008
4. Domain-Specific Modeling, Steven Kelly, Juha-Pekka Tolvanen, Wiley-Blackwell, 2008