
Ano Letivo 2019-20

Unidade Curricular TÓPICOS AVANÇADOS EM ENGENHARIA DE SOFTWARE

Cursos ENGENHARIA INFORMÁTICA (2.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14741036

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Línguas de Aprendizagem Inglês

Modalidade de ensino Presencial

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	30T; 30PL

* 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	30T; 30PL	168	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 (MDSO), sendo uma abordagem direcionada 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 MDSO;
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)

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

Os conteúdos programáticos estão em coerência com os objetivos da unidade curricular dado que o programa foi concebido para abordar de forma integrada o desenvolvimento baseado em modelos, introduzindo a terminologia e abordagens baseadas em modelos. No que respeita a aplicação de técnicas de transformação de modelos, vários temas foram selecionados, nomeadamente: meta-modelação, transformação de modelos, linguagens de domínio específico e ferramentas de MDD. Em relação aos vários objetivos da unidade curricular, ao longo da exposição dos conteúdos serão feitas alusões aos principais desafios que ocupam os analistas, programadores e investigadores na área de engenharia de software.

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:

- 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

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

As metodologias de ensino estão em coerência com os objetivos da unidade curricular dado que: 1) a exposição do programa associada à apresentação de casos práticos e à resolução de exercícios possibilita uma explicitação adequada dos conteúdos face ao público-alvo; 2) a exposição de evidência científica em conjunto com a análise de casos práticos permitem mostrar a importância do desenvolvimento baseado em modelos; 3) a exposição dos problemas atuais, complementadas com a realização de um trabalho prático possibilita a compreensão do conceito de desenvolvimento de software baseado em modelos, independentemente da plataforma. O regime de avaliação foi concebido para medir até que ponto as competências foram desenvolvidas.

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 2019-20

Course unit ADVANCED TOPICS IN SOFTWARE ENGINEERING

Courses INFORMATICS ENGINEERING

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area CIÊNCIA DE COMPUTADORES

Acronym

Language of instruction English

Teaching/Learning modality Presential

Coordinating teacher Paula Cristina Negrão Ventura Martins

Teaching staff	Type	Classes	Hours (*)
Paula Cristina Negrão Ventura Martins	PL; T	T1; PL1	30T; 30PL

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

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
30	0	30	0	0	0	0	0	168

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 (MDSD) 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 MDSD 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

Demonstration of the syllabus coherence with the curricular unit's learning objectives

The syllabus is consistent with the objectives of the curricular unit since the syllabus was designed to address, in an integrated way, Model-Driven Development, introducing the terminology and Model-Driven Architecture. Regarding the application of model transformation techniques, several themes were selected, namely: meta-modelling, model transformation, domain-specific languages and MDD tools. Concerning the main objectives of the curricular unit, the most challenging issues that occupy the attention of analysts, programmers and researchers in the software engineering area will be mentioned throughout the presentation of the syllabus.

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

Demonstration of the coherence between the teaching methodologies and the learning outcomes

The teaching methodologies are consistent with the objectives of the curricular unit because: 1) the exposition of the syllabus associated with the presentation of practical cases and the resolution of exercises allows an adequate explanation of the contents over the target public; 2) the exposition of scientific evidence together with the analysis of practical cases allows to show the importance of Model-Driven Development; 3) the exposition of current challenges, complemented with a practical work allows understanding the concept of platform-independent model regarding software development. The assessment scheme was designed to measure the extent to which skills were developed.

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