
Ano Letivo 2018-19

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 (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 (30 horas):

- As noções teóricas serão dadas por método predominantemente expositivo, com projecção e explicação dos objectivos e conteúdos correspondentes a cada tema, acompanhado de debate, colocação e esclarecimento de dúvidas.

Aulas Práticas (30 horas):

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

A avaliação tem duas componentes:

50% Provas escritas (PE) + 50% Trabalho(s) prático(s) (TP)

As componentes são classificadas de 0-20 valores, com classificação mínima de 10 valores no TP e de 6 valores na PE. A nota final será a média das notas da parte escrita, com a nota do(s) trabalho(s) prático(s),

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 2018-19

Course unit ADVANCED TOPICS IN SOFTWARE ENGINEERING

Courses INFORMATICS ENGINEERING

Faculty / School Faculdade de Ciências e Tecnologia

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 (MDSO) aims to use domain-specific languages (DSL) to create models that express application structure and behaviour. At 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. Rationale for Model-Driven Software Development
2. Terminology
3. Model-Driven Architecture (MDA)
4. Meta-Modelling
 - Concepts
 - OMG 4 layers 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. MDSO Tools

Teaching methodologies (including evaluation)

Theoretical lessons (T=30 hours)

- 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 (TP = 30 hours)

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

The assessment has two components:

50% Written exams (WE) + 50% Evaluation Work(s) (EW)

The components are classified from 0-20 values, with a minimum grade of 10 values in the EW and 6 values in the WE. The final grade is the average grade of the written exam, with the evaluation work(s).

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