

# Ficha de Unidade Curricular

**Mestrado Integrado em Engenharia Eletrotécnica e de Computadores**  
**Laboratório de Programação**  
**Ocorrência de: 2016/2017**

[Página oficial](#)

## Informação geral

**Unidade curricular:** Laboratório de Programação

**Código:** EEC0030

**Curso:** MIEEC, 4º

**Ano letivo:** 2016/2017

**Semestre:** 1S

**Créditos:** 6 ECTS

**Horas/Semanas:** 1x2T, 3x2P

**Professores:** [João Correia Lopes](#) (Regente)

## Língua de ensino

*Suitable for English-speaking students*

## Objetivos

No contexto da especificação, desenvolvimento e manutenção de aplicações de software com interface gráfica em arquiteturas cliente/servidor, esta unidade curricular tem por objetivos:

- Promover a aquisição de conceitos, métodos e técnicas de Engenharia de Software e dotar os estudantes da capacidade de os aplicar na conceção e desenvolvimento de sistemas de software.
- Dotar os estudantes de conhecimentos práticos na utilização de ferramentas de desenvolvimento de software adequadas à metodologia a usar e que permitam o acompanhamento do desenvolvimento do produto durante todo o seu ciclo de vida, incluindo a depuração, teste e documentação de código na linguagem de programação Java.

## Competências e resultados de aprendizagem

Ao completar esta unidade curricular o estudante deve ser capaz de:

1. Identificar e documentar os requisitos de um Sistema de Software utilizando *user stories*
2. Descrever os casos de utilização utilizando UML
3. Realizar um protótipo de interface com o utilizador
4. Identificar e documentar os requisitos suplementares
5. Obter o modelo conceptual do domínio utilizando UML
6. Obter modelos de processos de negócio utilizando UML
7. Descrever a arquitetura utilizando UML

8. Validar a arquitetura através de um protótipo vertical
9. Modelar a estrutura de classes utilizando UML
10. Modelar o comportamento de objetos utilizando UML
11. Elaborar a Ajuda ao Utilizador
12. Elaborar o Manual de Instalação
13. Codificar classes em Java usando as API standard
14. Documentar o código Java utilizando Javadoc
15. Testar o código utilizando Junit
16. Tratar da persistência dos dados
17. Utilizar uma ferramenta de elaboração colaborativa de documentação
18. Utilizar um IDE (*Integrated Development Environment*) na manutenção do software
19. Utilizar um sistema de controlo de versões

## Programa

- Introdução à Engenharia de requisitos. Documentação de requisitos.
- Linguagem de modelação UML.
- Projeto de software orientado por objetos. Modelação de arquitetura, estrutura de classes e comportamento.
- Projeto de interfaces pessoa-computador.
- Codificação usando a linguagem de programação Java.
- Verificação, validação e teste de software.
- Manutenção de software. Controlo de versões.
- Persistência de dados em SQL.
- Gestão e planeamento de projetos.

## Modo de trabalho

Presencial

## Pré-requisitos

Conhecimentos em Programação.

## Bibliografia principal

- Scott Ambler, *The Object Primer*, Cambridge University Press, 3rd Edition, 2004, ISBN: 978-0-521-54018-6 ([Biblioteca](#))
- F. Mário Martins, *Projetos de POO em JAVA*, FCA - Editora Informática, julho de 2014, ISBN: 978-972-722-792-1 ([Biblioteca](#))

## Bibliografia complementar

- Russ Miles e Kim Hamilton, *Learning UML 2.0*, O'Reilly, 2006, ISBN: 978-0-596-00982-3 ([Biblioteca](#))
- Bruce Eckel, *Thinking in Java*, Prentice Hall, 4th Edition, 2006, ISBN: 0131-87248-6 ([Biblioteca](#))
- Mauro Nunes e Henrique O'Neill, *Fundamental de UML*, 3ª edição, FCA - Editora Informática, 2004, ISBN: 978-972-722-481-4 ([Biblioteca](#))
- Alberto Manuel Rodrigues da Silva e Carlos Alberto Escaleira Videira, *UML, metodologias e ferramentas*

- CASE, 2ª Edição, Volume 1, Maio 2005, Centro Atlântico Editora, ISBN: 989-615-009-5 ([Biblioteca](#))
- Henrique O'Neill, Mauro Nunes e Pedro Ramos, *Exercícios de UML*, FCA - Editora Informática, 2010, ISBN: 978-972-722-616-0 ([Biblioteca](#))

## Métodos de ensino

A unidade curricular tem **aulas tutoriais** (2 horas por semana) e **aulas laboratoriais** (2 horas por semana) em sala de computadores.

Os tempos letivos das aulas tutoriais são usados para descrever brevemente as várias fases de desenvolvimento de software e os artefactos associados e para apresentar e exemplificar a utilização das metodologias e ferramentas a usar no trabalho prático seguindo o respetivo guião. São ainda indicadas referências de leitura.

Nas aulas laboratoriais os estudantes trabalham, em grupos de 4 ou 5, num projeto de software.

## Software

- [Enterprise Architect](#) (Windows)
- [IDE NetBeans](#) (Linux, MAC, Windows)
- [Dokuwiki](#) (Web)
- [SVN](#), [GIT](#)

## Palavras-chave

Ciências Físicas > Ciência de computadores > Programação  
Ciências Físicas > Ciência de computadores > Programação > Engenharia de software

## Modo de avaliação

Avaliação distribuída sem exame final

## Componentes de avaliação e ocupação registadas

Descrição	Tipo	Tempo (horas)	Data de Conclusão
Participação presencial (estimativa)	Aulas	56	
P1: Protótipo de interfaces com o utilizador	Trabalho laboratorial ou de campo	8	10/10/2016
P2: Especificação de requisitos	Trabalho laboratorial ou de campo	6	24/10/2016
P3: Projeto de alto nível e Protótipo	Trabalho laboratorial ou de campo	20	14/11/2016
P4: Projeto detalhado	Trabalho laboratorial ou de campo	8	05/12/2016
P5: Produto	Trabalho laboratorial ou de campo	60	02/01/2017
P6: Apresentação do Produto	Participação presencial	1	02/01/2017
P7: Desempenho individual	Trabalho		
J1-J4: Avaliação Java	Exercício		
M1: Miniteste Java	Teste/Exame	1,5	24/10/2016

Descrição	Tipo	Tempo (horas)	Data de Conclusão
M2: Miniteste UML	Teste/Exame	1,5	14/11/2016
	Total:	162	

## Obtenção de frequência

O trabalho prático (PROJ) será avaliado através da documentação apresentada, da aplicação desenvolvida e do desempenho individual nas aulas (P1 a P7).

Os conceitos teóricos (IND) são avaliados através da resposta individual, sem consulta, a dois mini-testes (M1 e M2), e através da elaboração de exercícios práticos em Java (J1 a J4).

A aprovação na unidade curricular está condicionada à obtenção de 50% em cada um dos trabalhos de avaliação prática (P1 a P7 de PROJ) e de 40% na componente de resposta individual (IND).

Esta unidade curricular, dada a sua natureza laboratorial, não é passível de avaliação em momento único, pelo que não poderá ser substituída por exame de recurso.

## Cálculo da classificação final

A nota final será calculada usando a fórmula:  $NOTA = 80\% PROJ + 20\% IND$

sendo:

$$PROJ = ((2 * P1 + 3 * P2 + 4 * P3 + 2 * P4 + 8 * P5 + P6) / 20) + P7$$

$$IND = (J1 + J2 + J3 + J4 + 8 * M1 + 8 * M2) / 20$$

A classificação da componente prática (PROJ) pode variar de elemento para elemento do mesmo grupo em mais ou menos 2 valores (P7), com base na opinião dos docentes e na autoavaliação e heteroavaliação a realizar internamente em cada grupo.

## Provas e trabalhos especiais

Para além da demonstração do trabalho prático, pode ter lugar uma prova oral para alguns estudantes.

## Avaliação especial (TE, DA, ...)

A avaliação prática e teórica, realizada nas aulas laboratoriais durante o semestre de funcionamento da unidade curricular, é exigida a todos os estudantes, independentemente do regime de inscrição e da necessidade de obter avaliação de frequência.

Os trabalhadores estudantes e equivalentes dispensados das aulas devem, com periodicidade a combinar com os docentes, apresentar a evolução dos seus trabalhos, assim como devem fazer a apresentação destes, simultaneamente com os estudantes ordinários, e realizar as provas teóricas e práticas de avaliação individual previstas.

## Melhoria de classificação final/distribuída

A nota final pode ser melhorada apenas através da frequência de numa nova edição da unidade curricular.

— JCL

From:

<https://web.fe.up.pt/~jlopes/> - JCL

Permanent link:

<https://web.fe.up.pt/~jlopes/doku.php/teach/lpro/201617/ficha>

Last update: **09/09/2017 17:07**

