

## Ficha de Unidade Curricular

<b>Curso</b>	Licenciatura em Engenharia Informática e de Computadores							
<b>Designação UC</b>	Programação Orientada por Objetos							
<b>Área Científica</b>	IC			<b>Observações</b>				
<b>Ano</b>	1	<b>Semestre</b>	2	<b>Duração<sup>1</sup></b>	Semestral			
<b>ECTS</b>	6	<b>Horas de trabalho<sup>2</sup></b>	162	<b>Horas de contacto<sup>3</sup></b>	TP	T	P	PL
					12	43,5		12
<b>Docente Responsável</b>	Pedro Alexandre de Seia e Cunha Ribeiro Pereira							

**Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes).**  
(1.000 carateres).

Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:

1. Definir e usar classes derivadas e classes que representem estruturas de dados dinâmicas.
2. Usar interfaces gráficas nas aplicações construídas.
3. Construir aplicações simples usando o paradigma da Programação Orientada por Objetos.
4. Testar e corrigir aplicações, usando as ferramentas de desenvolvimento adequadas.
5. Escrever relatórios onde se justifica a hierarquia de classes e as estruturas de dados usadas nas aplicações construídas.

**Conteúdos programáticos (1.000 carateres).**

- I. Herança e polimorfismo: classes derivadas; classes abstratas; interfaces; enumerados; ligação dinâmica.
- II. Classes internas; classes locais; classes anónimas; expressões lambda.
- III. Estruturas de dados elementares dinâmicas: vetores e listas ligadas. Iteradores. Genéricos.
- IV. Coleções; *Streams* binários e de texto para escrita e leitura. Tratamento de exceções.
- V. Utilização de interfaces gráficas: programação *event-driven*; *listeners*; layouts; arquitetura MVC.

**Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular (1.000 carateres).**

Esta unidade curricular introduz os conceitos e o vocabulário fundamental dos paradigmas da programação orientada por objetos (I) e da programação *event-driven* (V), concretizados na linguagem Java (1), (3) e (4). A hierarquia de classes dos *streams* e a hierarquia de classes das estruturas de dados dinâmicas da *framework* de coleções do Java (IV) são usadas para consolidar os conceitos transmitidos.

**Metodologias de ensino (avaliação incluída) (1.000 carateres).**

Previstas 30 aulas durante o semestre (15 de 3 horas e 15 de 1,5 horas). As aulas destinam-se à apresentação dos temas e de exemplos práticos de aplicação. Os tópicos principais são ainda explorados através da realização de trabalhos práticos para desenvolver programas em Java usando interface gráfica.

Os resultados da aprendizagem (1) e (2) são avaliados individualmente através do teste final escrito e das fichas realizadas durante o semestre. Durante o acompanhamento dos trabalhos de grupo realizados nas aulas práticas são avaliados os resultados da aprendizagem (3) e (5). Os resultados da aprendizagem (4) e (5) são avaliados na discussão final dos trabalhos de grupo.

Nota final corresponde a 40% do teste final, 40% dos trabalhos e 20% das fichas realizadas nas aulas.

<sup>1</sup> Anual, semestral, trimestral, ...

<sup>2</sup> Número total de horas de trabalho

<sup>3</sup> Discriminadas por tipo de metodologia adotado (T - Ensino teórico; TP - Ensino teórico-prático; PL - Ensino prático e laboratorial; TC - Trabalho de campo; S - Seminário; E - Estágio; OT - Orientação tutorial; O - Outro)

## Ficha de Unidade Curricular

### Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular (3.000 carateres).

O conhecimento dos conceitos fundamentais da programação orientada por objetos e da programação *event-driven* é obtido através de aulas interativas e respetivos elementos de apoio, e da realização de uma aplicação, com a participação dos alunos, que necessita de especializar comportamentos (polimorfismo), reutilizar código (herança) e utilizar interface com o utilizador, textual ou gráfica, (1) a (4).

A competência para desenvolver boas práticas de desenho e de escrita de código para aplicações usando o paradigma da programação orientada por objetos e interfaces gráficas (3-6) é desenvolvida através da realização de trabalhos com supervisão e da sua avaliação crítica.

### Bibliografia de consulta/existência obrigatória (1.000 carateres).

W. Savitch, *Java: An Introduction to Problem Solving and Programming*, 8th Edition, Pearson, 2017. ISBN 9780134462035