

## Ficha da unidade curricular (versão A3ES)

### 6.2.1.1. Unidade curricular

Modelação e Padrões de Desenho / Modeling and Design Patterns

### 6.2.1.2. Docente responsável e respetivas horas de contacto na unidade curricular (preencher o nome completo)

Jorge Manuel Rodrigues Martins Pião , 67,5 horas

### 6.2.1.3. Outros docentes e respetivas horas de contacto na unidade curricular

### Other academic staff and lecturing load in the curricular unit

### 6.2.1.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes)

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

1. Descrever os padrões de desenho de software utilizados frequentemente.
2. Modelar soluções orientadas por objectos (OO) usando formalismos e diagramas UML.
3. Analisar e justificar a aplicação de um padrão de desenho de software num determinado contexto.
4. Aplicar testes unitários, como prática essencial no desenvolvimento de software;
5. Desenvolver programas extensíveis através da utilização da API de reflexão.

### Learning outcomes of the curricular unit

A student completing this course unit should be able to:

1. Describe the software design patterns most commonly used.
2. Modeling object oriented (OO) solutions using UML diagrams.
3. Analyze and explain the application of a software design pattern in a specific context.
4. Employ unit tests in software development as an essential practice.
5. Develop extensible programs through the reflection API.

### 6.2.1.5. Conteúdos programáticos

1. Introdução aos padrões de desenho de software OO;
2. Utilização de testes unitários, como prática essencial no desenvolvimento de software;
3. Java Object Model, com maior ênfase nos Genéricos e Anotações;
4. API de reflexão Java.
5. Inversão de controlo (IoC) e injeção de dependências (DI);
6. Padrões OO para acesso a dados relacionais.

### Syllabus

1. Introduction to the OO software design patterns.
2. Employing unit testing as a common essential practice in the software development.
3. The Java Object Model, focusing in Generics and Annotations.
4. The Java reflection API.
5. Inversion of control (IoC) and dependency injection (DI).
6. OO design patterns for relational data access.

### 6.2.1.6. Demonstração da coerência dos conteúdos programáticos com os objetivos da unidade curricular

Esta unidade curricular tem o objectivo principal de aprofundar os conceitos da programação Orientada por Objectos (OO). Descreve os diagramas usados no desenho de soluções OO, introduz os processos de desenvolvimento usados em engenharia de software e transmite a importância dos padrões de desenho de software como descrições esquemáticas de soluções OO para problemas recorrentes.

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

This course aims to deepen the main concepts of Object Oriented Programming (OO). It describes the main diagrams used in the design of object oriented solutions, introduces the development process employed in the software engineering and it presents the importance of the software design patterns as generic OO solutions to solve recurrent problems.

#### **6.2.1.7. Metodologias de ensino (avaliação incluída)**

A metodologia de ensino desenvolve-se em várias componentes:

T – 10 horas de contacto teóricas - Exposição e discussão dos conceitos teóricos, incentivando à interactividade e colocação de questões;

TP – 35 horas teórico-práticas: Por cada tema teórico são apresentados problemas e discutidas diferentes soluções com os alunos através da sua modelação no quadro. A solução final é implementada na aula.

PL – 22,5 horas de contacto de prática laboratorial: Os conceitos teóricos são aprofundados através da resolução de problemas individualmente por cada aluno.

Além de um projecto final, são realizadas 2 séries de exercícios ao longo do semestre acompanhadas de 2 aulas de laboratório para esclarecimento de dúvidas.

A avaliação final é composta por: 40% (Teste) + 40% (Projecto + Discussão) + 20% (Fichas).

São realizadas 3 fichas de avaliação durante o semestre, contando apenas as 2 melhores notas na avaliação final.

#### **Teaching methodologies (including evaluation)**

The teaching methodology is developed in several components:

T– 10 theoretical teaching contact hours: Presentation and discussion of theoretical concepts, interacting and promoting the discussion with the students.

TP - 35 theoretical and practical teaching contact hours: For every theoretical concept we present a corresponding problem and we discuss several solutions based in the students' opinions. The final solution is implemented in collaboration with the students;

PL - 22.5 laboratory practice contact hours: Theoretical concepts are further developed through the resolution of several problems individually by each student.

In addition to a final project, the students must resolve 2 sets of problems during the course and followed by two labs for inquiries.

The final grade is composed by: 40% (Exam) + 40% (Project + Discussion) + 20% (Short exam).

The students will have 3 short exams during the semester and counting only the top 2 grades in the final evaluation.

#### **6.2.1.8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular**

Os temas são introduzidos através da apresentação de problemas que são debatidos com os alunos, sob orientação do docente e solucionados através do desenvolvimento de modelos no quadro, que traduzem as sugestões da audiência. Por fim é implementada a solução que melhor responde aos requisitos de eficiência, flexibilidade e extensibilidade.

As séries de exercícios pretendem replicar o mesmo tipo de problemas apresentados nas aulas, mas aplicados num novo contexto, devendo ser resolvidos autonomamente pelo aluno e com o apoio do docente no esclarecimento de eventuais dúvidas.

O trabalho final reúne um conjunto de desafios que deverão ser solucionados através da aplicação dos conhecimentos adquiridos do ponto 1 a 5. A avaliação deste projecto é baseada numa discussão onde são discutidas as soluções desenvolvidas pelo aluno.

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

The concepts are introduced through the presentation of issues that are discussed with the students, under the teacher guidance and solved through the development of models in the whiteboard, which reflect the suggestions from the audience. Finally we implemented the solution that best meets the requirements of efficiency, flexibility and extensibility.

The series of exercises intended to replicate the same kind of problems presented in class, but applied in a new context, and should be resolved independently by the student and with the support of teachers in clarifying any doubts.

The final work includes a set of challenges that must be solved through the application of knowledge gained from point 1 to 5. The evaluation of this project is based on a discussion where we discuss the solutions developed by the student.

#### **6.2.1.9. Bibliografia principal**

Cay S. Horstmann, Object-Oriented Design and Patterns, Wiley; 2 edition, June 2005

Martin Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, Nov 2002