

---

## 1. Caracterização da Unidade Curricular

### 1.1 Designação

[3114] Modelação e Padrões de Desenho / Modeling and Design Patterns

### 1.2 Sigla da área científica em que se insere

IC

### 1.3 Duração

Unidade Curricular Semestral

### 1.4 Horas de trabalho

160h 00m

### 1.5 Horas de contacto

Total: 67h 30m das quais T: 45h 00m | TP: 22h 30m

### 1.6 ECTS

6

### 1.7 Observações

Unidade Curricular Obrigatória

---

## 2. Docente responsável

[697] Jorge Manuel Rodrigues Martins Pião

---

## 3. Docentes e respetivas cargas letivas na unidade curricular

Não existem docentes definidos para esta unidade curricular

---

## 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.

**4. Intended learning outcomes  
(knowledge, skills and  
competences to be developed  
by the students)**

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.
- 

**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.
- 

**5. 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. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem 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.

---

**6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes**

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.

---

**7. Metodologias de ensino (avaliação incluída)**

A avaliação é distribuída com exame final. A avaliação tem as componentes escrita (E) e prática (P). A componente E é realizada através de exame escrito, com o mínimo de 9,5 valores.

A componente P, é baseada em três trabalhos práticos todos pedagogicamente fundamentais, realizados em grupo. Cada trabalho tem a nota mínima de 9,5 valores.

O conhecimento de cada estudante é avaliado durante a discussão dos trabalhos, sendo atribuídas classificações individuais (mínimo de 9,5 val).

A nota final (C) é obtida através da fórmula:  $C = E*0,5 + P*0,5$ .

---

**7. Teaching methodologies (including assessment)**

The assessment is distributed with a final exam. The assessment has written (E) and practical (P) components. Component E is carried out through a written exam, with a minimum of 9.5 points.

The P component is based on three practical works, all pedagogically fundamental, carried out in groups. Each work has a minimum grade of 9.5.

Each student's knowledge is assessed during the discussion of the work, with individual classifications being given (minimum of 9.5 val).

The final grade (C) is obtained using the formula:  $C = E*0.5 + P*0.5$ .

---

**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. Os trabalhos práticos 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. A avaliação dos trabalhos é baseada numa discussão onde são discutidas as soluções desenvolvidas pelo aluno.

---

**8. Evidence of the teaching methodologies coherence with the curricular unit's intended 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 practical exercises are 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 evaluation of the practical exercises is based on a discussion where we discuss the solutions developed by the student.

---

**9. Bibliografia de consulta/existência obrigatória**

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

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

---

**10. Data de aprovação em CTC** 2024-07-17

---

**11. Data de aprovação em CP** 2024-06-26