
1. Designação da unidade curricular

[4447] Modelação e Desenvolvimento de Software / Software Modelling and Development

2. Sigla da área científica em que se insere

IC

3. Duração

Unidade Curricular Semestral

4. Horas de trabalho

162h 00m

5. Horas de contacto

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

6. % Horas de contacto a distância

Sem horas de contacto à distância

7. ECTS

6

8. Docente responsável e respetiva carga letiva na Unidade Curricular

[697] Jorge Manuel Rodrigues Martins Pião | Horas Previstas: N/D

9. Outros docentes e respetivas cargas letivas na unidade curricular

Não existem docentes definidos para esta unidade curricular

10. Objetivos de aprendizagem e a sua compatibilidade com o método de ensino (conhecimentos, aptidões e competências a desenvolver pelos estudantes).

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

1. Elencar os principais padrões de desenho de software orientado por objetos.
2. Tirar partido dos mecanismos de reflexão/introspeção na construção de programas genéricos ou extensíveis.
3. Usar abstrações de ordem superior que representam os efeitos, ou o estado de controlo de um programa, tais como: *streams*, futuros, promessas e continuações.
4. Desenvolver componentes de software que disponibilizam operações assíncronas com base em abordagens não bloqueantes.

10. Intended Learning objectives and their compatibility with the teaching method (knowledge, skills and competences by the students).

A student completing this course unit should be able to:

1. Describe the main design patterns for object-oriented software.
2. Leverage reflection/introspection mechanisms in building generic or extensible programs.
3. Use higher-order abstractions that represent the effects or control flow of a program, such as streams, futures, promises, and continuations.
4. Develop software components providing asynchronous operations based on non-blocking approaches.

11. Conteúdos programáticos

- I. Introdução aos padrões de desenho de software orientados por objetos (OO);
- II. Mecanismos de reflexão/introspeção nos ambientes virtuais de execução.
- III. Aspectos funcionais em linguagens de programação; funções de 1ª classe (e.g., expressões lambda); funções de ordem superior e composição de funções; versão funcional de padrões de desenho OO.
- IV. Construção e utilização de sequências lazy. Interfaces genéricas para streams e iteração.
- V. Modelos de programação assíncrona e reactiva.
- VI. Utilização de APIs assíncronas; futuros, promessas e continuações; *streams* reactivas.

11. Syllabus

I. Introduction to object-oriented (OO) software design patterns
II. Reflection/introspection mechanisms in virtual execution environments.
III. Functional aspects in programming languages; first-class functions (e.g., lambda expressions); higher-order functions and function composition; functional version of OO design patterns.
IV. Construction and use of lazy sequences. Generic interfaces for streams and iteration.
V. Asynchronous and reactive programming models.
Vi. Use of asynchronous APIs; futures, promises and continuation; reactive streams.

12. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular

Esta unidade curricular aprofunda os conceitos da programação Orientada por Objectos (OO) e da construção de programas extensíveis (I, II) através da apresentação e utilização dos padrões de desenho mais importantes e das técnicas de reflexão/introspeção (1,2). A utilização de técnicas da programação funcional (III, IV) na construção e utilização de sequências lazy (4). Os conhecimentos de programação assíncrona (3,4) são transmitidos em (V) e através da utilização de construções e técnicas explicadas em (VI).

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

This curricular unit deepens the concepts of Object Oriented Programming (OO) and the construction of extensible programs (I, II) through the presentation and use of the most important design patterns and reflection/introspection techniques (1,2). The use of functional programming techniques (III, IV) in the construction and use of lazy sequences (4). Knowledge of asynchronous programming (3,4) is conveyed in (V) and through the use of constructs and techniques explained in (VI)

13. Metodologias de ensino e de aprendizagem específicas da unidade curricular articuladas com o modelo pedagógico

Aulas teóricas, com apresentação dos temas, exposição dos conceitos e exemplos de aplicação, complementadas com aulas de prática laboratorial, organizados em grupos, para realização de exercícios de consolidação e de trabalhos práticos.

13. Teaching and learning methodologies specific to the curricular unit articulated with the pedagogical model

Theoretical classes, with presentation of themes, exposition of concepts and application examples, followed by laboratory practice, organized in groups of students, with consolidation exercises, and practical assignments.

14. Avaliação

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: $C = E*0,5 + P*0,5$.

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

15. 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 que refletem as sugestões da audiência. Por fim, é implementada a solução que melhor responde aos requisitos de eficiência, flexibilidade e extensibilidade. Os trabalhos práticos, desenvolvidos em grupo, pretendem replicar o mesmo tipo de problemas apresentados nas aulas, mas aplicados num novo contexto, devendo ser resolvidos autonomamente com o apoio do docente no esclarecimento de eventuais dúvidas. A avaliação dos trabalhos é baseada em discussão, em que são discutidas as soluções desenvolvidas pelo grupo.

15. Evidence of the teaching methodologies coherence with the curricular unit's intended learning outcomes

The topics are introduced through the presentation of problems that are discussed with the students, under instructor guidance, and solved through the development of models that reflect the audience's suggestions. Finally, the solution that best meets the requirements of efficiency, flexibility, and extensibility is implemented. Practical work, developed in groups, aims to replicate the same types of problems presented in class but applied in a new context, and should be solved independently with the instructor's support in clarifying any doubts. Work assessment is based on discussions where the solutions developed by each group are analyzed.

16. Bibliografia de
consulta/existência obrigatória

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

R. Urma, M. Fusco, A. Mycroft, *Modern Java in Action - Lambdas, streams, functional and reactive programming*, Manning Publications, 2018.

17. Observações

Unidade Curricular Obrigatória

Data de aprovação em CTC:

Data de aprovação em CP: