

4.2.1. Designação da unidade curricular (PT):

Algoritmia e Programação

4.2.1. Designação da unidade curricular (EN):

Algorithmics and Programming

4.2.2. Sigla da área científica em que se insere (PT):

CI

4.2.2. Sigla da área científica em que se insere (EN):

CS

4.2.3. Duração (anual, semestral ou trimestral) (PT):

Semestral

4.2.3. Duração (anual, semestral ou trimestral) (EN):

Semiannual

4.2.4. Horas de trabalho (número total de horas de trabalho):

162.0

4.2.5. Horas de contacto:

Presencial (P) - TP-15.0; PL-45.0

4.2.6. % Horas de contacto a distância:

0.00%

4.2.7. Créditos ECTS:

6.0

4.2.8. Docente responsável e respetiva carga letiva na Unidade Curricular:

• Fátima Manuela da Silva Leal - 60.0h

4.2.9. Outros docentes e respetivas cargas letivas na unidade curricular:

[sem resposta]

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

A UC "Algoritmia e Programação" constitui a introdução aos conhecimentos e competências na área de programação.

Os estudantes que completarem a UC serão capazes de:

- i. Reconhecer e explicar o papel dos algoritmos na resolução de problemas.
- ii. Conhecer e saber aplicar os conceitos de programação estruturada, programa, estrutura de dados e algoritmo.
- iii. Criar programas de computador de pequena dimensão que demonstrem os conceitos, princípios e métodos de programação.
- iv. Interpretar e modificar programas de computador de dimensão média cujo estilo de escrita seja clara.

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

The CU "Algorithmics and Programming" constitutes the introduction to knowledge and skills in the area of programming.

Students who complete the CU will be able to:

- i. Recognize and explain the role of algorithms in problem solving.
- ii. Know and apply the concepts of structured programming, program, data structure and algorithm.
- iii. Create small computer programs that demonstrate programming concepts, principles and methods.
- iv. Interpret and modify medium-sized computer programs whose writing style is clear.

4.2.11. Conteúdos programáticos (PT):

1. Visão geral e história do computador e das linguagens de programação
2. Conceitos básicos
 - 2.1 Programação estruturada
 - 2.2 Notação utilizada
 - 2.3 Operadores
3. Algoritmia
 - 3.1 Pseudocódigo e flowcharts
 - 3.2 Instruções
 - 3.2.1 Sequenciais
 - 3.2.2 Decisão
 - 3.2.3 Repetição
4. Traçagem e Teste
5. Introdução à linguagem Python
 - 5.1 Tipos de dados
 - 5.2 Condições e ciclos
 - 5.3 Strings listas e dicionários

4.2.11. Conteúdos programáticos (EN):

1. Overview and history of the computer and programming languages
2. Basic concepts
 - 2.1 Structured programming
 - 2.2 Notation
 - 2.3 Operators
3. Algorithms
 - 3.1 Pseudocode and flowcharts
 - 3.2 Instructions
 - 3.2.1 Sequential
 - 3.2.2 Decision
 - 3.2.3 Repetition
4. Tracing and Testing
5. Introduction to Python language
 - 5.1 Data types
 - 5.2 Conditions and cycles
 - 5.3 Strings lists and dictionaries

4.2.12. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular. (PT):

Os conteúdos programáticos lecionados permitem aos estudantes adquirir conhecimentos relevantes com os quais serão capazes de perceber a complexidade do desenho de um algoritmo e como implementar um programa. Nomeadamente os conteúdos 1 a 6 permitem ao aluno atingir os objetivos (i) e (ii). Os conteúdos 7 a 8 contribuem para os objetivos (iii) e (iv). O conjunto integrado dos conteúdos contribui para que o aluno saiba desenhar, implementar e testar um programa.

4.2.12. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular. (EN):

The syllabus allows students to acquire relevant knowledge with which they will be able to understand the complexity of designing an algorithm and how to implement a program. Namely, contents 1-6 allow the student to achieve objectives (i)-(ii). Contents 7-8 contribute to objectives (iii)-(iv). The integrated set of contents helps the student know how to design, implement and test a program.

4.2.13. Metodologias de ensino e de aprendizagem específicas da unidade curricular articuladas com o modelo pedagógico. (PT):

O modelo de ensino será totalmente presencial contendo aulas de dois tipos: (i) teórico-práticas com exposição, demonstração e discussão dos temas abordados; e (ii) práticas para realização de trabalhos práticos para a resolução de pequenos programas com vista a aplicarem os conceitos ensinados. Serão utilizadas como plataformas de suporte o Colibri-Zoom e Moodle.

4.2.13. Metodologias de ensino e de aprendizagem específicas da unidade curricular articuladas com o modelo pedagógico. (EN):

The teaching model will be completely face-to-face, containing two types of classes: (i) theoretical-practical with exposure, demonstration and discussion of the topics covered; and (ii) practices for carrying out practical work to solve small programs in order to apply the concepts taught. Colibri-Zoom and Moodle will be used as support platforms.

4.2.14. Avaliação (PT):

Será utilizado o método de avaliação contínua, considerando 2 provas práticas (P1 e P2) com o peso de 35% cada e 1 projeto (TP) com o peso de 30%. Em caso de reprovação ou melhoria de nota, será considerado um exame final com a ponderação de 100%.

4.2.14. Avaliação (EN):

The assessment will be continuous, consisting of 2 practical tests (P1 and P2) weighing 35% each and 1 project (TP1) weighing 30%. In case of failure or grade improvement, a final exam will be considered with a weighting of 100%.

4.2.15. Demonstração da coerência das metodologias de ensino e avaliação com os objetivos de aprendizagem da unidade curricular. (PT):

A união entre a exposição teórica da matéria e a apresentação de exemplos práticos permite aos alunos a familiarização com os conceitos fundamentais e a linguagem de programação e atingir os objetivos (i)-(ii). A resolução de problemas simples guiada será utilizada para atingir os objetivos (iii)-(iv).

4.2.15. Demonstração da coerência das metodologias de ensino e avaliação com os objetivos de aprendizagem da unidade curricular. (EN):

The union between the theoretical exposition of the subject and the presentation of practical examples allows students to familiarize themselves with the fundamental concepts and the programming language and achieve objectives (i)-(ii). Simple guided problem solving will be used to achieve objectives (iii)-(iv).

4.2.16. Bibliografia de consulta/existência obrigatória (PT):

Adelaide Carvalho, Práticas de Python: algoritmia e programação FCA, 2021, 978-972-722-918-5.

Luciano Ramalho, Fluent Python: Clear, Concise, and Effective Programming, O'Reilly Media, 1st edition, 2015, 978-1491946008.

Cormen, T., Leiserson, C., Rivest, R., Stein, C., Introduction to Algorithms, MIT Press; 3rd edition, 2009.

4.2.16. Bibliografia de consulta/existência obrigatória (EN):

Adelaide Carvalho, Práticas de Python: algoritmia e programação FCA, 2021, 978-972-722-918-5.

Luciano Ramalho, Fluent Python: Clear, Concise, and Effective Programming, O'Reilly Media, 1st edition, 2015, 978-1491946008.

Cormen, T., Leiserson, C., Rivest, R., Stein, C., Introduction to Algorithms, MIT Press; 3rd edition, 2009.

4.2.17. Observações (PT):

[sem resposta]

4.2.17. Observações (EN):

[sem resposta]