
Ano Letivo 2020-21

Unidade Curricular INTRODUÇÃO À PROGRAMAÇÃO CIENTÍFICA

Cursos BIOENGENHARIA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 19071001

Área Científica ENGENHARIA ELETRÓNICA

Sigla

Línguas de Aprendizagem Português

Modalidade de ensino Presencial

Docente Responsável Maria Margarida da Cruz Silva Andrade Madeira e Carvalho de Moura

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Maria Margarida da Cruz Silva Andrade Madeira e Carvalho de Moura	PL; T	T1; PL1	14T; 42PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
1º	S1	14T; 42PL	156	6

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Esta cadeira não requer conhecimentos de programação, mas apenas familiaridade com sistemas informáticos.

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

No final desta disciplina, os estudantes aprovados terão demonstrado ser capazes de:

- Compreender a utilização da programação na resolução de problemas do âmbito da sua disciplina de estudos.
- Identificar as principais componentes dos sistemas computacionais e sua relação com a programação.
- Dominar com razoável à-vontade as técnicas de programação elementares.
- Resolver autonomamente problemas de programação simples.
- Apreçar a complexidade algorítmica dos programas que escrevem.
- Conhecer os fundamentos da programação vetorial e matricial usando NumPy.
- Tirar partido das funcionalidades de programação numérica de SciPy.
- Tirar partido das técnicas básicas da programação gráfica (como representação simples de pontos em 2D e 3D, histogramas, gráficos de barras) usando Matplotlib.
- Utilizar funcionalidades de bibliotecas de estatística, álgebra simbólica e manipulação de dados (como SymPy e Pandas).

Conteúdos programáticos

Introdução à Programação

Elementos constituintes dos programas

Algoritmos básicos

Estruturas de dados fundamentais

Funções

Recursividade

Iteração

Asserções

Buscas e ordenações

Classes e Programação orientada pelos objetos

Desenvolvimento de soluções usando bibliotecas

Metodologias de ensino (avaliação incluída)

Nas aulas teóricas, o professor discute os temas da disciplina, usando apresentações de suporte, para fazer demonstrações e para ilustrar o desenvolvimento de programas. Nas aulas práticas, os alunos resolvem pequenos problemas de programação ou realizam trabalhos mais longos, com guião, no computador. Os alunos completarão a sua formação através de trabalho individual ou em grupo, realizado fora das aulas.

A avaliação usa a modalidade *avaliação por frequência*, nos termos do artigo 9, n.º 1, alínea b) do Regulamento Geral de Avaliação da Universidade do Algarve, de 31 de agosto de 2016. Não há exame.

Bibliografia principal

John V- Guttag, *Introduction to Computation and Programming Using Python*, ISBN-13: 978-0262525008, ISBN-10: 0262525003, MIT Press, 2016.

Ernesto Costa, *Programação em Python - Fundamentos e Resolução de Problemas*, ISBN: 978-972-722-816-4, FCA, 2015.

Joey Bernard, *Python Recipes Handbook: A Problem-Solution Approach*, Apress, 2016, SBN 978-1-4842-0241-8, <https://www.amazon.com/Python-Recipes-Handbook-Problem-Solution-Approach/dp/1484202422>

Academic Year 2020-21

Course unit INTRODUCTION TO SCIENTIFIC PROGRAMMING

Courses BIOENGINEERING

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area

Acronym

Language of instruction
Portuguese

Teaching/Learning modality
Lectures and labs.

Coordinating teacher Maria Margarida da Cruz Silva Andrade Madeira e Carvalho de Moura

Teaching staff	Type	Classes	Hours (*)
Maria Margarida da Cruz Silva Andrade Madeira e Carvalho de Moura	PL; T	T1; PL1	14T; 42PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
14	0	42	0	0	0	0	0	156

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

This course unit does not require programming knowledge, just familiarity with computer systems.

The students intended learning outcomes (knowledge, skills and competences)

By the end of this unit, successful students will have demonstrated being able to:

- Understand the use of programming in the resolution of problems in the scope of the field of study.
- Identify the main components of computer systems and its relation to programming.
- Master with relative ease the elementary programming techniques.
- Individually solve simple programming problems.
- Assess the algorithmic complexity of the programs produced.
- Know the principles of vector and matrix programming usando NumPy.
- Take advantage of the numerical programming features of SciPy.
- Take advantage of the basic techniques of graphical programming (such as simple representation of points in 2D and 3D, histograms, bar charts) using Matplotlib.
- Use libraries for statistics, symbolic algebra, and data manipulation (such as SymPy and Pandas).

Syllabus

Introduction to programming

Programming building elements

Basic algorithms

Fundamental data structures

Functions

Recursivity

Iteration

Assertions

Search and sort orders

Classes and object oriented programming

Developing solutions using libraries

Teaching methodologies (including evaluation)

In the lectures, the teacher discusses the topic of the course, using his computer to present the course slides, to make experiments and demonstrations, and to illustrate the development of programs.

In the labs, students solve small problems programming or perform longer programming assignments. Students complete their training through individual or group work, done outside the classroom.

The evaluation uses the modality of *evaluation by frequency*, as prescribed in the general regulations of the university. There is no exam.

Main Bibliography

John V. Guttag, *Introduction to Computation and Programming Using Python*, ISBN-13: 978-0262525008, ISBN-10: 0262525003, MIT Press, 2016.

Ernesto Costa, *Programação em Python - Fundamentos e Resolução de Problemas*, ISBN: 978-972-722-816-4, FCA, 2015.

Joey Bernard, *Python Recipes Handbook: A Problem-Solution Approach*, Apress, 2016, SBN 978-1-4842-0241-8, <https://www.amazon.com/Python-Recipes-Handbook-Problem-Solution-Approach/dp/1484202422>