Ficha de Unidade Curricular A3ES
Programação
Licenciatura em Engenharia Biomédica
2025-26

1. Designação da unidade curricular

[4046] Programação / Computer Programming

2. Sigla da área científica em que se insere	INF
3. Duração	Unidade Curricular Semestral
4. Horas de trabalho	150h 00m
5. Horas de contacto	Total: 62h 00m das quais T: 30h 00m | TP: 30h 00m | O: 2h 00m
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	[1501] Nelson Guerreiro Cortez Nunes | Horas Previstas: 120 horas
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).

1. Conhecer os objetivos da programação e sua utilização num contexto de Engenharia Biomédica.

2. Conhecer diversos tipos de variáveis e aprender a manipulá-las.

3. Saber utilizar diversas funções elementares e estruturas de decisão e repetição.

4. Aprender a desenvolver programas de forma estruturada.

5. Ter contacto com ferramentas informáticas para a programação e desenvolvimento de algoritmos de apoio à Engenharia Biomédica.

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

1. Identify computer programming purposes and its use in Biomedical Engineering context.

2. Recognize different types of variables and learn how to manipulate them.

3. Identify how to use several elementary functions, decision, and repetition structures.

4. Learn how to develop programs in a structured way.

5. Contact with programming computer tools and algorithm development for Biomedical Engineering.

11. Conteúdos programáticos

1. Introdução histórica à computação; Unidades de processamento e a estrutura de comunicação; Sistemas operativos; Linguagens de programação.

2. Algoritmia. Conceitos teóricos sobre algoritmia. Pseudo-linguagem e fluxogramas.

3. Tipos de dados e variáveis.

4. Expressões aritméticas e lógicas.

5. Estruturas de controlo (sequencias, condições e ciclos).

6. Programação em Python.

6.1. O IDE. Strings, listas, tuplos e dicionários. Manipulação de variáveis com indexação.

6.2. Estruturas de controlo/recursividade em Python.

6.3. Funções e módulos.

6.4. Ficheiros.

6.5. Bibliotecas externas.

6.5. Construção de ambientes gráficos.

11. Syllabus

1. Historical perspectives of computing; processing units and communication structure; Operating systems; Programming languages.

2.Algorithms. Theoretical concepts about algorithms, Pseudo-language, and flowcharts.

3. Data types and variables.

4. Arithmetic and logical expressions.

5. Control structures (sequences, conditions, and cycles).

6. Python programming. The IDE.

6.1. Strings, lists, tuples, and dictionaries. Manipulation of variables with indexing.

6.2. Control/recursion structures in Python.

6.3. Functions and modules.

6.4. Files.

6.5. External libraries.

6.6. Graphical environments.

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

O primeiro capítulo permite aos alunos conhecerem a necessidade de se terem desenvolvido linguagens de programação como plataformas de cálculo automatizado. É também focada a ligação do hardware (dispositivos I/O, processador e memória) com o software (entradas, saídas, processamento lógico e aritmético e armazenamento de dados).

No segundo capítulo são lecionados conceitos fundamentais de algoritmia, dando a conhecer os diversos tipos de dados bem como as principais estruturas de programação. Este capítulo desafia os alunos a estruturar o seu pensamento, desenvolvendo algoritmos que resolvam problemas computacionais.

A aplicabilidade da algoritmia é feita na linguagem de programação Python, onde algoritmos sobre problemas aplicados à engenharia biomédica são implementados. Para uma correta aplicação do algoritmo na linguagem de programação, os alunos necessitam de escrever as instruções respeitando a sintaxe e sequenciar as instruções de forma lógica e coerente.

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

The first chapter allows students to learn about the need to have developed programming languages as automated calculation platforms. It is also focused on connecting the hardware (I/O devices, processor, and memory) with the software (inputs, outputs, logical and arithmetic processing, and data storage).

In the second chapter, fundamental concepts of algorithms are taught, making known the various types of data as well as the main programming structures. This chapter challenges students to structure their thinking by developing algorithms that solve computational problems.

The applicability of algorithm is performed in the Python programming language, where algorithms applied to biomedical engineering problems are implemented. For a correct application of the algorithm in the programming language, students need to write the instructions respecting the syntax and sequence the instructions logically and consistently.

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

A unidade curricular está dividida em aulas teóricas e práticas, sendo 50% das aulas lecionadas em sala de aula e 50% no laboratório.

A componente teórica é lecionada através de exposição oral , apresentação de casos de estudos e disponibilização de material didático complementar.

A componente prática é estruturada em sessões de programação guiada, exercícios individuais e em grupo . É utilizada, ainda, a revisão de código.

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

The curricular unit is divided into theoretical and practical classes, with 50% of the classes delivered in the classroom and 50% in the computer lab.

The theoretical component is delivered through oral presentations, case study presentations, and the provision of supplementary teaching materials.

The practical component is structured around guided programming sessions, individual and group exercises, and includes code review."

14. Avaliação

A avaliação de conhecimentos é feita através de um projeto (NP), trabalho pedagogicamente fundamental, e de um teste (NT) ou um exame final (NE).

A nota final é calculada pela seguinte equação: NF = 0.6*(NT ou NE) + 0.4*NP.

Condições essenciais para aprovação:

Em teste ou exame, obter uma nota mínima de 9.50 valores.

Realizar a entrega do projeto e apresentação dentro do prazo estipulado e obter nota mínima de 9.50 valores.

14. Assessment

The evaluation is accomplished through a project (NP), pedagogically fundamental work, and a test (NT) or a final exam (NE).

The final score is calculated by the following equation: NF = 0.6*(NT or NE) + 0.4*NP.

Indispensable conditions for approval:

One test or exam, with a minimum score of 9.50.

Presentation and delivery of the project within the stipulated time with a minimum score of 9.50 grade.

15. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

As aulas são divididas em teóricas e práticas, sendo 50% lecionadas em sala de aula e 50% no laboratório. As aulas teóricas alternam com aulas práticas que são realizadas no Laboratório de Informática, permitindo que os alunos pratiquem na plataforma computacional

Após a introdução dos conceitos fundamentais e sua implementação na plataforma computacional são realizados diversos problemas onde os alunos terão de estruturar o seu raciocínio, elaborar algoritmos e a implementá-los recorrendo à sintaxe da linguagem de programação.

O projeto permite aos alunos trabalhar em equipa no desenvolvimento de um algoritmo computacional com aplicação na área da Engenharia Biomédica, desde a análise do problema, estruturação dos dados e elaboração do algoritmo até à implementação numa linguagem de programação.

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

Classes are divided into theoretical and practical, being 50% taught in the classroom and 50% in the laboratory. The theoretical classes alternate with practical classes that are held in the Computer Laboratory, allowing students to practice on the computer platform.

After the introduction of the fundamental concepts and their implementation in the computational platform are realized several problems where students will have to structure their reasoning, elaborate algorithms, and implement them using the syntax of the programming language.

The project allows students to work as a team in the development of a computational algorithm with application in Biomedical Engineering, from problem analysis, data structuring and algorithm elaboration to implementation in a programming language.

16. Bibliografia de consulta/existência obrigatória

[1] Martins, J.P. (2015) Programação em Python ? Introdução à programação utilizando múltiplos paradigmas, Lisboa, IST Press.

[2] Heys, J.J. (2017) Chemical and Biomedical Engineering Calculations Using Python, USA, John Wiley & Sons.

[3] Costa, E. (2015) Programação em Python, Fundamentos e resolução de problemas, Lisboa, FCA.

[4] Schneider, D.I. (2016) An Introduction to Programming Using Python, USA, Pearson

[5] Guttag, J. (2016) Introduction to Computation and Programming Using Python with Application to Understanding Data, Cambridge, MA, MIT Press

17. Observações

Unidade Curricular Obrigatória

Data de aprovação em CTC: 2024-07-17

Data de aprovação em CP: 2024-06-26