

## Ficha de Unidade Curricular – (Versão A3ES 2018-2023)

### 1. Caracterização da Unidade Curricular.

- 1.1. **Designação da unidade curricular** (1.000 carateres).  
Programação Orientada para Objetos / Object Oriented Programming
- 1.2. **Sigla da área científica em que se insere** (100 carateres).  
CE
- 1.3. **Duração**<sup>1</sup> (100 carateres).  
Semestral
- 1.4. **Horas de trabalho**<sup>2</sup> (100 carateres).  
162 h
- 1.5. **Horas de contacto**<sup>3</sup> (100 carateres).  
22.5 T, 22.5 TP, 22.5 PL
- 1.6. **ECTS** (100 carateres).  
6
- 1.7. **Observações**<sup>4</sup> (1.000 carateres).
- 1.7. **Remarks** (1.000 carateres).

### 2. Docente responsável e respetiva carga letiva na Unidade Curricular (preencher o nome completo) (1.000 carateres). João Beza Teixeira Seixas e Sousa, 67.5 horas.

### 3. Outros docentes e respetivas cargas letivas na unidade curricular (1.000 carateres).

### 4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes). (1.000 carateres).

1. Utilização da linguagem de programação Python, na sua vertente de programação orientada por objetos.
2. Desenvolvimento de uma aplicação usando programação orientada por objetos. A aplicação em causa deverá usar entidades matemáticas cuja implementação tire partido da programação orientada por objetos. Um exemplo de uma aplicação nestas condições é um "Ray Tracer". Um "Ray Tracer" serve para obter imagens de cenas 3D, e usa entidades matemáticas como pontos, vetores, matrizes, retas e planos, cuja implementação natural assenta no encapsulamento de dados e operações através de programação orientada por objetos.

### 4. Intended learning outcomes (knowledge, skills and competences to be developed by the students). (1.000 characters).

1. Use of the Python programming language, in its object-oriented programming paradigm.
2. Development of an application using object-oriented programming. The application in question should use mathematical entities whose implementation takes advantage of object-oriented programming. An example of an application in these conditions is a "Ray Tracer". A "Ray Tracer" is used to obtain images of 3D scenes, and uses mathematical entities such as points, vectors, matrices, lines and planes, whose natural implementation is based on the encapsulation of data and operations through object-oriented programming.

### 5. Conteúdos programáticos (1.000 carateres).

Classes, objetos, atributos de dados – membros, atributos funções – métodos, encapsulamento, construtor, self, redefinição de operadores, isinstance, herança, classe base, classe derivada, super, polimorfismo, exceções.

### 5. Syllabus (1.000 characters).

Classes, objects, data attributes - members, function attributes - methods, encapsulation, constructor, self, operator overloading, isinstance, inheritance, base class, derived class, super, polymorphism, exceptions.

### 6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade

**curricular (1.000 caracteres).**

Esta unidade curricular aprofunda a programação de computadores fazendo a transição da programação procedimental para a programação orientada por objetos, através da linguagem de programação Python, na sua vertente de programação orientada por objetos. Os conteúdos programáticos constituem uma introdução à programação orientada por objetos em Python. Os objetivos de aprendizagem estão fortemente relacionados entre si e são abrangidos pela totalidade dos conteúdos programáticos.

**6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes (1.000 characters).**

This course deepens computer programming by making the transition from procedural programming to object-oriented programming, through the Python programming language, in its object-oriented programming paradigm. The syllabus is an introduction to object-oriented programming in Python. The learning objectives are strongly related to each other and are covered by the entire syllabus.

**7. Metodologias de ensino (avaliação incluída) (1.000 caracteres).**

Ensino teórico-prático. As aulas compreendem exposição dos conceitos teóricos (T), resolução de exercícios (TP) e programação de computador (PL). Existem trabalhos realizados individualmente ao longo do semestre fora das aulas. Existe um projeto final a realizar individualmente fora das aulas.

Os conceitos teóricos, a resolução de exercícios e a programação de computador são avaliados em exame (NE). Os conceitos teóricos, a resolução de exercícios e a programação de computador são usados e avaliados nos trabalhos realizados individualmente ao longo do semestre (NT). A programação de computador é usada no projeto final e é avaliada numa discussão individual (NP). A nota final (NF) é obtida por  $NF=0.33*NE+0.33*NT+0.34*NP$ . Para obter aprovação é preciso ter nota mínima de 9.5 valores em NE, NP e NF.

**7. Teaching methodologies (including assessment) (1.000 characters).**

Theoretical-practical teaching. Classes include exposure of theoretical concepts (T), exercise resolution (TP) and computer programming (PL). There are assignments done individually throughout the semester outside of classes. There is a final project to be carried out individually outside of classes.

Theoretical concepts, exercise resolution and computer programming are evaluated in exam (NE). Theoretical concepts, exercise resolution and computer programming are used and evaluated in the assignments carried out individually throughout the semester (NT). Computer programming is used in the final project and is evaluated in an individual discussion (NP). The final grade (NF) is obtained by  $NF = 0.33 * NE + 0.33 * NT + 0.34 * NP$ . To obtain approval it is necessary to have a minimum grade of 9.5 values in NE, NP and NF.

**8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular (3.000 caracteres).**

O objetivo de aprendizagem 1 é suportado pelas aulas teóricas, pelas aulas de resolução de exercícios, pelas aulas de programação e pelos trabalhos realizados ao longo do semestre.

O objetivo de aprendizagem 2 é suportado pela realização do projeto.

**8. Evidence of the teaching methodologies coherence with the curricular unit's intended learning outcomes (3.000 characters).**

Learning outcome 1 is supported by theoretical classes, exercise solving classes, programming classes and assignments carried out throughout the semester.

Learning objective 2 is supported by the project work.

**9. Bibliografia de consulta/existência obrigatória (1.000 caracteres).**

Swaroop C. H., A Byte of Python. (Livro livre disponível em <https://python.swaroopch.com/>)

Allen B. Downey, Think Python, 2nd Edition, Green Tea Press, 2015. (Livro livre disponível em <https://greenteapress.com/wp/think-python-2e/>)

Joel Grus, Data Science from Scratch: First Principles with Python, 2nd Edition. O'Reilly, 2019.

<sup>1</sup> Anual, semestral, trimestral, ...

<sup>2</sup> Número total de horas de trabalho.

<sup>3</sup> Discriminadas por tipo de metodologia adotado (T - Ensino teórico; TP - Ensino teórico-prático; PL - Ensino prático e laboratorial; TC - Trabalho de campo; S - Seminário; E - Estágio; OT - Orientação tutorial; O - Outro).

<sup>4</sup> Assinalar sempre que a unidade curricular seja optativa.