

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

### 1. Caracterização da Unidade Curricular.

- 1.1. **Designação da unidade curricular** (1.000 carateres).  
Programação Orientada por 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 Bezeira 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 aplicações, recorrendo ao paradigma da programação orientada por objetos e ao encapsulamento de dados.
3. Adquirir conhecimento sobre as construções da linguagem Python, na sua vertente de programação orientada por objetos.
4. Definir e utilizar classes e classes derivadas e outros conceitos relacionados com objetos.
5. Conceber e escrever programas que resolvam problemas recorrendo ao paradigma da programação orientada por objetos.
6. Produzir documentação técnica para os programas desenvolvidos.
7. Utilizar de forma adequada as ferramentas de desenvolvimento e de produção de documentação.

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

1. Use the Python programming language, in its object-oriented programming paradigm.
2. Application development, using the object-oriented programming paradigm and data encapsulation.
3. Acquire knowledge about the constructions of the Python language, in its object-oriented programming paradigm.
4. Define and use classes and derived classes and other concepts related to objects.
5. Design and write programs that solve problems using the object-oriented programming paradigm.
6. Produce technical documentation for the developed programs.
7. Properly use development and documentation production tools.

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

1. Classes, objetos, atributos de dados – membros, atributos funções – métodos, encapsulamento.
2. Construtor, “self”.
3. Redefinição de operadores.
4. “isinstance”, herança, classe base, classe derivada, super, polimorfismo.
5. Exceções.

**5. Syllabus (1.000 characters).**

1. Classes, objects, data attributes - members, function attributes - methods, encapsulation.
2. Constructor, “self”.
3. Operator overloading.
4. “isinstance”, inheritance, base class, derived class, super, polymorphism.
5. Exceptions.

**6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular (1.000 carateres).**

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 outcomes are strongly related to each other and are covered by the entire syllabus.

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

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. Os conceitos teóricos, a resolução de exercícios e a programação de computador são avaliados em exame (NE) e nos trabalhos realizados individualmente ao longo do semestre (NT). A nota final (NF) é obtida por  $NF=0,5*NE+0,5*NT$ . Para obter aprovação é preciso ter nota mínima de 9,5 valores em NE e NF. Os alunos poderão ser submetidos a uma prova complementar para validação das classificações obtidas.

**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. Theoretical concepts, exercise resolution and computer programming are evaluated in exam (NE) and in the assignments carried out individually throughout the semester (NT). The final grade (NF) is obtained by  $NF=0,5*NE+0,5*NT$ . To obtain approval it is necessary to have a minimum grade of 9,5 values in NE and NF. Students may be submitted to an extra assessment to validate the obtained classifications.

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

Os objetivos de aprendizagem 1 a 7 são suportado pelas aulas teóricas, pelas aulas de resolução de exercícios, pelas aulas de programação e pelos trabalhos realizados ao longo do semestre.

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

Learning outcomes 1 to 7 are supported by theoretical classes, exercise solving classes, programming classes and assignments carried out throughout the semester.

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

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.