

#### **Caracterização da Unidade Curricular / Characterization of the Curricular Unit**

**Designação da Unidade Curricular (UC) / Title of Curricular Unit (CU):** Algoritmia e Técnicas de Programação / Algorithms and Programming Techniques

**Área científica da UC / CU Scientific Area:** Ciências Informáticas / Computer Science

**Semestre / Semester:** 1º

**Número de créditos ECTS / Number of ECTS credits:** 6

**Carga horária por tipologia de horas / Workload by type of hours:** TP: 45; OT: 6; O: 9

**Carga letiva semanal / Weekly letive charge:** 3h

#### **Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):**

Pretende-se que, no final do semestre, os alunos sejam capazes de:

- Estruturar um programa e reconhecer os principais conceitos associados a essa prática;
- Compreender e praticar a utilização de uma linguagem de programação.

#### **Intended learning outcomes (knowledge, skills and competences to be developed by the students):**

It is intended that at the end of the semester, students are able to:

- Design a program and recognize the key concepts associated with this practice;
- Understanding and practice the use of a programming language.

#### **Conteúdos programáticos:**

##### **1. Introdução à algoritmia**

- 1.1. Definição de algoritmo
- 1.2. Representação de algoritmos

##### **2. Conceitos de programação**

- 2.1. Conceção de um programa
- 2.2. Execução de um programa
- 2.3. Compiladores e interpretadores
- 2.4. Linguagens de programação

##### **3. Programação estruturada**

- 3.1. Tipos de dados, variáveis e constantes
- 3.2. Operadores e expressões
- 3.3. Instruções de controlo de fluxo

- 3.4. Sub-algoritmos
- 3.5. Variáveis indexadas

#### **4. Exemplificação de algoritmos em Java, C#, Python**

##### **Syllabus:**

##### **1. Introduction to Algorithms**

- 1.1. Algorithms Definition
- 1.2. Algorithms Representation

##### **2. Programming Concepts**

- 2.1. Designing a program
- 2.2. Implementation of a program
- 2.3. Compilers and interpreters
- 2.4. Programming Languages

##### **3. Structured Programming**

- 3.1. Types of data variables and constants
- 3.2. Operators and Expressions
- 3.3. Instructions Flow Control
- 3.4. Sub-algorithms
- 3.5. Indexed variables

##### **4. Exemplification of algorithms in Java, C#, Python**

---

