

---

**Ano Letivo** 2019-20

---

**Unidade Curricular** PROGRAMAÇÃO

---

**Cursos** SISTEMAS E TECNOLOGIAS DE INFORMAÇÃO

---

**Unidade Orgânica** Instituto Superior de Engenharia

---

**Código da Unidade Curricular** 18121001

---

**Área Científica** FORMAÇÃO TÉCNICA, CIÊNCIAS INFORMÁTICAS

---

**Sigla** FT

---

**Línguas de Aprendizagem** Português

---

**Modalidade de ensino** Presencial

---

**Docente Responsável** Pedro Jorge Sequeira Cardoso

---

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
João Miguel Fernandes Rodrigues	PL; TP	TP1; PL1	15TP; 45PL

\* 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	15TP; 45PL	168	6

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

#### Precedências

Sem precedências

#### Conhecimentos Prévios recomendados

--

#### Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Apresentar as técnicas algorítmicas de resolução de problemas usando uma primeira abordagem à programação, aplicando o paradigma procedimental, para posteriormente ser introduzida a programação orientada a objetos.

Iniciar os alunos na análise, técnicas de formalização, resolução de problemas tipificados e codificação numa linguagem de programação.

#### Conteúdos programáticos

1. Algoritmia e técnicas de programação.
2. Introdução às linguagens de programação
  - Estrutura de um programa;
  - Representação da informação;
  - Tipos de dados Primitivos;
  - Tipos de dados Referências;
  - Constantes e variáveis;
  - Expressões;
  - Conversões;
  - Operadores;
  - Classes da interface de programação.
3. Entrada e saída formatadas de dados.
4. Estruturas de controlo (decisórias e repetitivas).
5. Programação Procedimental.
6. Sequências/Arrays.
7. Sequências de Caracteres.
8. Ficheiros.

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

O encadeamento e a sequência dos conteúdos programáticos permitirão desenvolver uma aprendizagem gradual, geradora das seguintes competências que estão de acordo com os objetivos pre-estabelecidos:

- 1-Capacidade de resolver problemas recorrendo a técnicas algorítmicas;
- 2-Capacidade de saber codificar numa linguagem de programação problemas com diversos mecanismos de controlo, tipos e estruturas de dados.

### **Metodologias de ensino (avaliação incluída)**

#### Métodos de Ensino Aprendizagem

Aulas TP: Método expositivo, com projeção e explicação dos objetivos e conteúdos correspondentes a cada semana.

Aulas P: Prática laboratorial em computador. Resolução e codificação de problemas.

#### Modo de Avaliação

A avaliação compreende duas componentes: um teste ou exame (TE) e trabalhos práticos (TPs). As duas componentes são avaliadas na escala de 0 a 20. A classificação final é igual a 20% (T/E) + 80% (TPs), com classificação mínima de 7 valores em cada componente. O aluno fica aprovado se obtiver classificação final igual ou superior a 10.

De acordo com o n.º 3 do artigo 6.º do despacho reitoral RT 59/2015, de 28 de julho, nos cursos técnicos superiores profissionais, a inclusão do cumprimento do dever de assiduidade nos métodos de avaliação é obrigatória, nos seguintes termos:

a) Considera-se que um estudante cumpre o dever de assiduidade a uma UC, quando não exceda o número limite de faltas correspondente a 25% das horas de contacto previstas

---

### **Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular**

A metodologia é baseada em aulas interativas com os alunos. São inicialmente introduzidos alguns conceitos e problemas.

Posteriormente, os alunos resolvem os problemas, supervisionados pelo docente, aplicando e aprofundando os conceitos.

O recurso à plataforma de e-learning, as ferramentas de software e a prática laboratorial em computador são fundamentais na aprendizagem. O trabalho desenvolve-se individualmente e em grupo, servindo as necessidades de resolução dos problemas/projetos, sendo centrado no aluno e assumindo a diversidade de modelos pessoais de aprendizagem. Os alunos atingem assim os objetivos, acima propostos, ao trabalharem de acordo com esta metodologia.

---

### **Bibliografia principal**

- [1] Portal da UC de Programação - Tutoria eletrónica: <http://tutoria.ualg.pt>
- [2] Rocha, A. D., Pacheco, O. R.; Introdução à Programação em JAVA, FCA-Informática
- [3] Mendes, A. J., Marcelino, M. J.; Fundamentos de Programação em JAVA, FCA-Informática
- [4] Jesus, C.; Curso Prático de Java, FCA-Informática
- [5] Carvalho, A.; Exercícios de Java - Algoritmia e Programação Estruturada, FCA-Informática

**Academic Year** 2019-20

**Course unit** PROGRAMMING I

**Courses** SISTEMAS E TECNOLOGIAS DE INFORMAÇÃO

**Faculty / School** INSTITUTE OF ENGINEERING

**Main Scientific Area** FORMAÇÃO TÉCNICA, CIÊNCIAS INFORMÁTICAS

**Acronym** FT

**Language of instruction** Portuguese

**Teaching/Learning modality** Presential

**Coordinating teacher** Pedro Jorge Sequeira Cardoso

Teaching staff	Type	Classes	Hours (*)
João Miguel Fernandes Rodrigues	PL; TP	TP1; PL1	15TP; 45PL

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

#### Contact hours

T	TP	PL	TC	S	E	OT	O	Total
0	15	45	0	0	0	0	0	168

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

--

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

Present students to algorithmic techniques for problem solving using a first approach to programming, applying the procedural paradigm, in order to introduced later the object oriented programming paradigm.

Present students to the analysis, formalization techniques, problem solving and coding in a programming language.

#### Syllabus

1. Algorithm and programming techniques.
2. Introduction to programming languages
  - Structure of a program;
  - Representation of information;
  - Primitive data types;
  - References;
  - Constants and variables;
  - Expressions;
  - Conversions;
  - Operators;
  - Programming interface classes
3. Format input and output data.
4. Control structures (decision-making and repetitive).
5. Procedural programming.
6. Arrays.
7. Strings
8. Files.

#### Demonstration of the syllabus coherence with the curricular unit's learning objectives

The chaining and sequencing of the syllabus contents will allow the development of a gradual learning, generating the following competences that are in agreement with the pre-established objectives:

- 1-Ability to solve problems using algorithmic techniques;
- 2-Ability to code in a programming language problems with various control mechanisms, types and data structures.

### Teaching methodologies (including evaluation)

#### Teaching Methods Learning

TP classes: expository method, with projection and explanation of the objectives and contents corresponding to each week.

P classes: Laboratory practice in computer. Resolution and codification of problems.

#### Evaluation Mode

The evaluation comprises two components: a test or exam (TE) and practical assignments (TP). The two components are evaluated on the scale of 0 to 20. The final classification is equal to 20% TE + 80% TP, with a minimum classification of 7 values in each component. The student is approved if he or she has a final grade of 10 or higher.

Pursuant to n.o 3 of Article no. 6 of RT 59/2015, of July 28, in the professional higher technical courses, the inclusion of the fulfillment of the duty of assiduity in the methods of evaluation is obligatory, in the following terms:

a) It is considered that a student fulfills the duty of assiduity to a CU, when it does not exceed the limit of absences corresponding to 25% of the foreseen contact hours.

---

### Demonstration of the coherence between the teaching methodologies and the learning outcomes

The methodology is based on interactive classes with students. Some concepts and problems are initially introduced. Later, the students solve the problems, supervised by the teacher, applying and deepening the concepts.

The use of the e-learning platform, software tools and computer laboratory practice are fundamental in learning. The work is developed individually and in groups, serving the needs of solving problems / projects, being student-centered and assuming the diversity of personal learning models.

---

### Main Bibliography

- [1] Portal da UC de Programação - Tutoria eletrónica: <http://tutoria.ualg.pt>
- [2] Rocha, A. D., Pacheco, O. R.; Introdução à Programação em JAVA, FCA-Informática
- [3] Mendes, A. J., Marcelino, M. J.; Fundamentos de Programação em JAVA, FCA-Informática
- [4] Jesus, C.; Curso Prático de Java, FCA-Informática
- [5] Carvalho, A.; Exercícios de Java - Algoritmia e Programação Estruturada, FCA-Informática