

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

1. Caracterização da Unidade Curricular.

1.1. **Designação da unidade curricular** (1.000 carateres).
Programação I/Programming I

1.2. **Sigla da área científica em que se insere** (100 carateres).
IC

1.3. **Duração**¹ (100 carateres).
Semestral

1.4. **Horas de trabalho**² (100 carateres).
162h

1.5. **Horas de contacto**³ (100 carateres).
67,5h (T – 25h; TP - 20h; PL- 22,5h)

1.6. **ECTS** (100 carateres).
6

1.7. **Observações**⁴ (1.000 carateres).
UC comum com outros cursos

1.7. **Remarks** (1.000 carateres).
UC offered in other courses

2. **Docente responsável e respetiva carga letiva na Unidade Curricular** (preencher o nome completo) (1.000 carateres).
Jorge Manuel Rodrigues Martins Pião (67,5h)

3. **Outros docentes e respetivas cargas letivas na unidade curricular** (1.000 carateres).
Manuel Fernandes Carvalho (202,5h); Diogo Sérgio Esteves Cardoso (67,5h)

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

Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:

1. Demonstrar o conhecimento sobre os mecanismos básicos das linguagens procedimentais.
2. Construir programas de pequena complexidade em linguagem C a partir da sua descrição em linguagem natural.
3. Usar ferramentas para construir, testar e corrigir pequenos programas.

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

Students who successfully complete this course will be able to:

1. Demonstrate knowledge of the basic mechanisms of procedural languages.
2. Build small complexity programs in C language from its natural language description.
3. Use tools to build, test and debug small programs.

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

- I. Linguagens de programação versus linguagens naturais. Léxico, sintaxe e semântica. Algoritmos. A linguagem de programação C.

- II. Tipos de dados elementares. Valores, operadores e expressões. Operações de entrada/saída de dados. Variáveis.
- III. Instruções de controlo de fluxo: decisão simples e múltipla; ciclos. Funções; Passagem de parâmetros; Visibilidade e tempo de vida de variáveis.
- IV. Tipos estruturados: *arrays*; *strings*; estruturas; *arrays* multidimensionais
- V. Algoritmos básicos de pesquisa e ordenação.
- VI. Introdução aos ponteiros. Acesso a ficheiros.
- VII. Introdução à programação modular.

5. Syllabus (1.000 characters).

- I. Programming languages versus natural languages. Lexicon, syntax and semantics. Algorithms. The C programming language.
- II. Basic data types. Values, operators and expressions. Input /output operations. Variables.
- III. Flow control instructions: simple and multiple decision; cycles. Functions; Passing parameters; Visibility and lifetime of variables.
- IV. Structured types: arrays; strings; structures; multidimensional arrays.
- V. Basic algorithms for searching and sorting.
- VI. Introduction to pointers. Accessing files.
- VII. Introduction to modular programming.

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

Esta unidade curricular representa para a maioria dos alunos um primeiro contacto com a programação, que se pretende motivador sem descurar o formalismo e o rigor, sendo a base da formação de software do curso. São introduzidos conceitos e vocabulário fundamental da programação procedimental (pontos I, II, III e IV do conteúdo programático) concretizados na linguagem C. Os algoritmos de pesquisa e ordenação e o acesso a ficheiros (V, VI) são usados para praticar os conceitos transmitidos. O conceito de ponteiro é a introdução à programação modular (pontos VI e VII) preparam discussões posteriores na área da programação e da arquitetura de computadores.

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

This course is for most students a first contact with programming, with motivational purposes, without neglecting formalism and rigor, being the basis of software training course. The fundamental concepts and vocabulary of procedural programming (Sections I, II, III and IV of the syllabus) are presented using the C language. Searching and sorting algorithms and access to files aim to practice the transmitted concepts. Pointers and an introduction to modular programming are introduced in (section VI and VII) prepare subsequent discussions in the area of programming and computer architecture.

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

Ensino teórico-prático, estando previstas 30 aulas durante o semestre a que correspondem 67,5 horas de contacto (15 aulas de 3 horas e 15 de 1,5 horas). O tempo total de trabalho do estudante é de 160 horas. As aulas destinam-se à apresentação dos temas e de exemplos práticos de aplicação. Os tópicos principais são ainda explorados através da realização de trabalhos práticos (TP). Os resultados da aprendizagem são avaliados individualmente através do teste escrito (TE) e das fichas (FI) realizadas durante o semestre, e na discussão final (DF) dos trabalhos de grupo. A nota final é obtida pela fórmula $(TP + DF) * 0,2 + TE * 0,4 + FI * 0,2$. As componentes TP, DF e TE têm de ser iguais ou superiores a 10 valores e a componente FI igual ou superior a 8 valores.

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

Theoretical and practical teaching is planned during the semester in 30 lectures that correspond to 67.5 of contact hours (15 lessons of 3 hours and 15 of 1.5 hours). The total student working hours is 160. The lectures are intended for presentation of topics and practical examples. The main topics are further explored through practical work (PW).

Learning outcomes are assessed individually through a written test (WT) and assessment forms (AF) during the semester and in the final discussion (FD) of group work. Final grade is the result of the formula $(PW + FD) * 0.2 + WT * 0.4 + AF * 0.2$, where PW, FD and WT must be equal or greater than 10 and AF must be greater than 8 (in a scale of 0 to 20).

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

Os conteúdos programáticos são expostos nas aulas teórico-práticas, complementando uma apresentação interativa das matérias com a realização, pelos estudantes, de pequenos exercícios de consolidação das mesmas.

Os objetivos de aprendizagem são desenvolvidos na realização dos trabalhos de grupo.

São efetuadas aulas práticas de acompanhamento dos trabalhos de grupo, que complementam a avaliação, na discussão final, da globalidade dos objetivos de aprendizagem.

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

The course contents are exposed in practical classes, complementing an interactive presentation of the materials with the realization, by the students, of small consolidation exercises.

Learning outcomes are developed in the work group.

Lectures are also dedicating to monitoring practical group work, complementing the evaluation of overall learning outcomes on the final discussion.

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

K.N.King, *C Programming: a Modern Approach*, 2Ed., W. W. Norton & Company, 2008, ISBN 9780393979503

B. Kernighan, D. Ritchie, *The C Programming Language*, 2ª ed., Prentice Hall, 1988, ISBN 9780131103627.

¹ Anual, semestral, trimestral, ...

² Número total de horas de trabalho.

³ 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).

⁴ Assinalar sempre que a unidade curricular seja optativa.