
Ano Letivo 2021-22

Unidade Curricular PROGRAMAÇÃO IMPERATIVA

Cursos ENGENHARIA INFORMÁTICA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 140064322

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Código CNAEF (3 dígitos) 481

Contributo para os Objetivos de Desenvolvimento Sustentável - ODS (Indicar até 3 objetivos) 4, 8, 10

Línguas de Aprendizagem Português-PT

Modalidade de ensino

Presencial

Docente Responsável

Pedro João Valente Dias Guerreiro

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Pedro João Valente Dias Guerreiro	PL; T	T1; PL1; PL2	28T; 56PL
Maria da Graça Cristo dos Santos Lopes Ruano	PL	PL3; PL4	56PL
António Eduardo de Barros Ruano	PL	PL5	28PL
Joel David Valente Guerreiro	PL	PL6	28PL

* 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	28T; 28PL	156	6

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

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Programação Imperativa é a primeira cadeira de programação da licenciatura em engenharia informática. Está preparada no pressuposto que os alunos não têm conhecimentos de programação, ainda que reconhecendo que alguns possam ter.

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

No final da cadeira, os alunos deverão ser capazes de:

- Analisar e explicar o comportamento de programas escritos em C.
- Modificar e fazer evoluir programas escritos em C.
- Desenhar, implementar, testar e depurar programas escritos em C, usando as técnicas da programação imperativa.
- Aplicar a decomposição funcional na resolução de problemas de programação.
- Reconhecer a importância de seleccionar o algoritmo apropriado para cada tarefa de programação.
- Dominar os algoritmos básicos de busca e ordenação, as técnicas elementares do processamento de cadeias de caracteres e a operação com tabelas chave-valor.
- Tirar partido da linguagem de programação C para melhor explorar a arquitetura dos computadores na programação.
- Compreender o processo de desenvolvimento de software.

Conteúdos programáticos

1. Conceitos fundamentais: programação, linguagens de programação.
2. Programação em C: ambientes de desenvolvimento.
3. Decomposição funcional.
4. Programação de funções em C.
5. Tipos de dados em C.
6. Estruturas de controlo.
7. Funções recursivas.
8. Arrays.
9. Algoritmos de busca e ordenação.
10. Entradas e saídas.
11. Gestão dinâmica da memória.
12. Mecanismo de execução dos programas: pilha de execução.
13. Cadeias de caracteres.
14. Tabelas chave-valor.

Metodologias de ensino (avaliação incluída)

Nas aulas teóricas, o professor discute os temas da cadeira, usando o seu computador para exibir os transparentes, para fazer demonstrações e para ilustrar o desenvolvimento de programas.

Nas aulas práticas, os alunos resolvem pequenos problemas de programação ou realizam trabalhos mais longos, com guião, no computador.

Os alunos completarão a sua formação através de trabalho individual ou em grupo, realizado fora das aulas.

A avaliação usa a modalidade de avaliação por frequência, nos termos do art. 9, n. 1, al. b) do Regulamento Geral de Avaliação da Universidade do Algarve, de 31 de agosto de 2016. O exame assume a forma de uma prova escrita. Não são admitidos a exame os alunos cuja nota na componente prática seja inferior a um valor fixado no início de cada edição da cadeira (normalmente 7.5).

Bibliografia principal

- The C Programming Language, Dennis M. Richie, Brian W. Kernighan, <http://www.amazon.co.uk/C-Programming-Language-2nd/dp/0131103628/> .
- Elementos de Programação com C, Pedro Guerreiro, <http://www.fca.pt/pt/catalogo/informatica/programacao/elementos-de-programacao-com-c/> .
- Sebenta, formada pela coleção das apresentações usadas pelo professor nas aulas teóricas.

Academic Year 2021-22

Course unit IMPERATIVE PROGRAMMING

Courses INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area

Acronym

CNAEF code (3 digits) 481

Contribution to Sustainable Development Goals - SGD (Designate up to 3 objectives) 4, 8, 11

Language of instruction Portuguese-PT

Teaching/Learning modality Lectures and labs.

Coordinating teacher Pedro João Valente Dias Guerreiro

Teaching staff	Type	Classes	Hours (*)
Pedro João Valente Dias Guerreiro	PL; T	T1; PL1; PL2	28T; 56PL
Maria da Graça Cristo dos Santos Lopes Ruano	PL	PL3; PL4	56PL
António Eduardo de Barros Ruano	PL	PL5	28PL
Joel David Valente Guerreiro	PL	PL6	28PL

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

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
28	0	28	0	0	0	0	0	156

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

Imperative Programming is the first course on programming at the *licenciatura em engenharia informática* . It is prepared on the assumption that students do not have any prior knowledge on programming, while acknowledging that some may do.

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

At the end of the cours, students should be able to:

- Analyze and explain the behavior of programs written in C.
- Modify and evolve programs written in C.
- Design, implement, test and debug programs written in C, using the techniques of imperative programming.
- Apply functional decomposition to solve programming problems.
- Recognize the importance of selecting the appropriate algorithm for each programming task.
- Master the basic algorithms for searching and sorting, the elementary techniques of the strings processing and the operation of key-value tables.
- Take advantage of the C programming language to better exploit the architecture of computer programming.
- Understand the process of software development.

Syllabus

1. Fundamental concepts: programming, programming languages.
 2. Programming in C: development environments.
 3. Functional decomposition.
 4. Programming functions in C.
 5. C data types.
 6. Control structures.
 7. Recursive functions.
 8. Arrays.
 9. Algorithms for sorting and searching
 10. Inputs and output.
 11. Dynamic memory management.
 12. Program execution: the runtime stack.
 13. Strings.
 14. Key-value tables.
-

Teaching methodologies (including evaluation)

In the lectures, the teacher discusses the topic of the course, using his computer to present the course slides, to make experiments and demonstrations, and to illustrate the development of programs.

In the labs, students solve small problems programming or perform longer programming assignments.

Students complete their training through individual or group work, done outside the classroom.

The evaluation uses the modality of "evaluation by frequency", as prescribed in the general regulations of the university. The examination takes the form of a written test.

Main Bibliography

1. The C Programming Language, Dennis M. Richie, Brian W. Kernighan, <http://www.amazon.co.uk/C-Programming-Language-2nd/dp/0131103628/> .
2. Elementos de Programação com C, Pedro Guerreiro, <http://www.fca.pt/pt/catalogo/informatica/programacao/elementos-de-programacao-com-c/> .
3. Lecture slides, provided by the course staff.