

---

**Ano Letivo** 2018-19

---

**Unidade Curricular** ALGORITMOS E ESTRUTURA DE DADOS

---

**Cursos** ENGENHARIA ELÉTRICA E ELETRÓNICA (1.º ciclo)  
- RAMO DE TECNOLOGIAS DE INFORMAÇÃO E TELECOMUNICAÇÕES (1.º ciclo)

---

**Unidade Orgânica** Instituto Superior de Engenharia

---

**Código da Unidade Curricular** 15241017

---

**Área Científica** INFORMÁTICA

---

**Sigla**

---

**Línguas de Aprendizagem** Português (PT)

---

**Modalidade de ensino** Obrigatória

---

**Docente Responsável** Roberto Célio Lau Lam

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Roberto Célio Lau Lam	OT; PL; TP	TP1; PL1; OT1	30TP; 30PL; 24OT

\* 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
2º	S1	30TP; 30PL; 20OT	140	5

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

#### Precedências

Sem precedências

#### Conhecimentos Prévios recomendados

Conhecimentos adquiridos na disciplina de Programação. Conhecimentos da Linguagem C. Capacidade de raciocínio indutivo e dedutivo.

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

Fornecer aos alunos: conhecimentos sobre metodologia de programação orientada por objectos. Ter a capacidade de construir programas nesta metodologia de programação. Saber conceber, analisar, e realizar, algoritmos lógicos para resolução de problemas de pequena, média dimensão.

Conhecer e saber utilizar a anotação descritiva da eficiência algorítmica.

Saber utilizar os algoritmos básicos de ordenação e pesquisa: Bubble sort, Shell sort, Quick sort e as pesquisas sequencial e binária.

Saber utilizar estruturas de armazenamento lineares: vectores, pilhas, filas e hierárquicas: árvores.

Conhecer estruturas de armazenamento híbridas nomeadamente: Tabelas de dispersão (Hash tables), grafos, matrizes esparsas.

Conhecer situações de utilização prática das estruturas de armazenamento híbridas: determinação do caminho mais curto, multiplicação de matrizes esparsas. No final da disciplina deverão estar aptos a dimensionar estruturas de armazenamento de dados, de pequena e média dimensão, eficientes.

### Conteúdos programáticos

1. Programação Orientada por Objectos. Formalização teórica, exemplificação com JAVA: Classes, objectos. Variáveis, funções membro (métodos, interfaces), construtores e destruidor, sobrecarga de funções membro. Atributos de acesso. Operadores, sobrecarga de operadores, Herança, Polimorfismo.
  2. JAVA (utilização memória dinâmica, ficheiros e interfaces gráficas do utilizador - GUI).
  3. Complexidade algorítmica.
  4. Algoritmos de ordenação em vectores: *Bubble sort*, *Shell sorte* *Quick sort*
  5. Algoritmos de pesquisa em vectores
  6. Listas, sob os conceitos LILO, FIFO (Filas e Pilhas).
  7. Árvores. Terminologia. BST árvores de pesquisa binária. Percursos: prefixo, infixo e pósfixo. Implementação. Arvores AVL
  8. Tabelas dispersão e grafos. Terminologia e exemplos de implementação.
  9. Utilização prática do conteúdo da disciplina em aplicações práticas: Caminho mais curto. Matrizes de adjacências esparsas.
- 

### Metodologias de ensino (avaliação incluída)

As aulas teóricas e teórico-práticas terão uma pequena exposição teórica dos conteúdos, apresentação de estruturas armazenamento e algoritmos base. No final das aulas teórico-práticas são apresentados casos práticos com problemas para resolver. A orientação tutorial incidirá na resolução de problemas propostos bem como no apoio às deficiências que os alunos apresentem. A plataforma da tutoria electrónica da UAAlg será utilizada, com os seguintes objectivos: a) Publicação dos materiais das aulas práticas; b) Publicação de avaliações; c) Publicação de avisos da disciplina; d) Criação de um espaço de comunicação (fórum de discussão) para esclarecimento de dúvidas e estímulo da comunicação docente /alunos e alunos/alunos. Avaliação: Teste/exame escrito (50% nota final) e apresentação dos trabalhos práticos feitos em programação (50% nota final). Para aprovação da UC os alunos terão de obter pelo menos 7 valores (0-20) no teste/exame e 7 valores (0-20) nos trabalhos.

---

### Bibliografia principal

- Cormen, T.H., Leiserson, C.E., Rivest, R. and Stein, C., Introduction to Algorithms, MIT Press.
- Sedgewick, R., Addison-Wesley Publishing Company, Inc.
- Eckels, B., Thinking in JAVA I e II, 3th Edition.
- Estruturas de Dados e Algoritmos em JAVA A. Adrego da Rocha, 2011, FCA.
- Rodrigues P., Pereira P. E Sousa M., Programação em C++ Conceitos básicos e Algoritmos. FCA.
- Guerreiro, P., Elementos de Programação com C, FCA Lidel. 3ª Edição.

**Academic Year** 2018-19

**Course unit** ALGORITHMS AND DATA STRUCTURE

**Courses** ELECTRIC AND ELECTRONICS ENGINEERING  
- RAMO DE TECNOLOGIAS DE INFORMAÇÃO E TELECOMUNICAÇÕES (1.º ciclo)

**Faculty / School** Instituto Superior de Engenharia

**Main Scientific Area** INFORMÁTICA

**Acronym**

**Language of instruction** Portuguese (PT)

**Teaching/Learning modality** Required

**Coordinating teacher** Roberto Célio Lau Lam

Teaching staff	Type	Classes	Hours (*)
Roberto Célio Lau Lam	OT; PL; TP	TP1; PL1; OT1	30TP; 30PL; 24OT

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

### Contact hours

T	TP	PL	TC	S	E	OT	O	Total
0	30	30	0	0	0	20	0	140

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

Knowledge obtained in programming. Knowledge of C Language. Ability of inductive and deductive reasoning.

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

Objectives:

Provide students with knowledge about: a) object-oriented programming, b) search and sort algorithms, c) linear (lists) and hierarchical (trees) data structures and d) hash tables, matrices and adjacency lists.

Skills:

a) object oriented programming, b) construct and use search and sort routines, c) program and use simple and complex data structures (lists and trees) and d) to use class libraries to develop applications.

### Syllabus

1 Object Oriented Programming; 2 Java; 3 Algorithmic complexity; 4 Sorting algorithms in vectors; 5 Search algorithms in vectors; 6 Lists, under the concepts LIFO and FIFO (queues, stacks); 7 Trees (Binary search trees and AVLs); 8 Hash tables; 8 Graphs; 9 Use of the contents of the course in practical applications;

### Teaching methodologies (including evaluation)

The lectures and practical classes will have a small theoretical exposition of content, presentation of storage structures and basic algorithms. At the end of the theoretical-practical classes, case studies are presented with problems to solve. The tutorials will focus on problem solving as well as support offered to students who have disabilities. The electronic platform of UAAlg will be used with the following objectives: a) Publication of resources for practical classes, b) Publication of assessments; c) Publication of notices; d) creating a space for communication (discussion forum) to clarify questions and stimulate communication between teacher/students and students/students. Test / Exam, written examination (50% final) and presentation of practical work done on schedule (50% final). To be approved the students must obtain at least seven points (in a scale of 0;20) in the test / exam and at least 7 points (in a scale of 0;20) the component of the programming work.

### **Main Bibliography**

Cormen, T.H., Leiserson, C.E., Rivest, R. and Stein, C., Introduction to Algorithms, MIT Press.

Sedgewick, R., Addison-Wesley Publishing Company, Inc.

Eckels, B., Thinking in JAVA I e II, 3th Edition.

Estruturas de Dados e Algoritmos em JAVA A. Adrego da Rocha, 2011, FCA.

Rodrigues P., Pereira P. E Sousa M., Programação em C++ Conceitos básicos e Algoritmos. FCA.

Guerreiro, P., Elementos de Programação com C, FCA Lidel. 3ª Edição