

Algoritmos e Estruturas de Dados

Código: L.EIC011

Sigla: AED

Áreas Científicas	
Classificação	Área Científica
OFICIAL	Engenharia Informática e Computação

Ocorrência: 2021/2022 - 1S 

Ativa? Sim

Página Web: <https://www.dcc.fc.up.pt/~pribeiro/aulas/aed2122/>

Unidade Responsável: Departamento de Engenharia Informática

Curso/CE Responsável: Licenciatura em Engenharia Informática e Computação

Ciclos de Estudo/Cursos

Sigla	Nº de Estudantes	Plano de Estudos	Anos Curriculares	Créditos UCN	Créditos ECTS	Horas de Contacto	Horas Totais
<u>L.EIC</u>	250	<u>Plano Oficial</u>	2	-	6	52	162

Docência - Responsabilidades

Docente	Responsabilidade
<u>Pedro Manuel Pinto Ribeiro</u>	Regente
<u>Ana Paula Cunha da Rocha</u>	Regente

Docência - Horas

Teóricas: 2,00

Teórico-Práticas: 2,00

Tipo	Docente	Turmas	Horas
<u>Teóricas</u>	Totais	2	4,00
	<u>Pedro Manuel Pinto Ribeiro</u>		2,00
	<u>Ana Paula Cunha da Rocha</u>		2,00
<u>Teórico-Práticas</u>	Totais	12	24,00
	<u>Filipa Marília Monteiro Ramos</u>		4,00
	<u>Pedro Manuel Pinto Ribeiro</u>		2,00
	<u>Ana Paula Cunha da Rocha</u>		4,00
	<u>António Manuel Ribeiro Pereira da Costa</u>		4,00

Sofia Cardoso Martins		4,00
Luciano Polónia Gonçalves Grácio		2,00
Rosaldo José Fernandes Rossetti		4,00

 A ficha foi alterada no dia 2021-10-02.

Campos alterados: Provas e trabalhos especiais, Fórmula de cálculo da classificação final

Língua de trabalho

Português

Objetivos

No final da unidade curricular, os estudantes deverão ser capazes de:

- analisar (e medir experimentalmente) a complexidade temporal e espacial de algoritmos;
- analisar (e testar experimentalmente) a correção de algoritmos simples;
- conhecer os principais algoritmos de pesquisa e ordenação em arrays e a sua complexidade;
- compreender o conceito de tipo abstrato de dados e saber organizar programas em torno desse conceito;
- conhecer as estruturas de dados fundamentais (e algoritmos associados e respetiva complexidade) usadas para implementar eficientemente tipos abstratos de dados comuns em bibliotecas de coleções;
- saber escolher coleções, estruturas de dados e algoritmos apropriados para resolver problemas práticos;
- escrever programas em C++ que implementam e usam as estruturas de dados e algoritmos fundamentais

Resultados de aprendizagem e competências

No final da unidade curricular, os estudantes deverão ter competência para: modelar problemas recorrendo ao paradigma da orientação por objetos; resolver problemas fazendo uso de tipos de dados abstratos e estruturas de dados simples lineares e não lineares.

Modo de trabalho

Presencial

Pré-requisitos (conhecimentos prévios) e co-requisitos (conhecimentos simultâneos)

O estudante deve possuir conhecimentos básicos de programação e linguagem C++.

Programa

Conceitos e técnicas básicas: complexidade temporal e espacial de algoritmos; tipos abstratos de dados; análise da correção de algoritmos.

Algoritmos de pesquisa e ordenação em arrays.

Estruturas de dados lineares e sua implementação: pilhas, filas e listas.

Estruturas de dados hierárquicos e sua implementação: árvores binárias; árvores binárias de pesquisa; árvores binárias equilibradas; árvores B.

Tabelas de dispersão e algoritmos de manipulação.

Filas de prioridade e *heaps* binários

Algoritmos básicos em grafos: tipos de grafos; representação; pesquisa em largura e em profundidade; ordenação topológica; ciclos; conectividade; caminho mais curto; árvore de expansão mínima.

Bibliografia Obrigatória

Weiss, Mark Allen; [Data structures and algorithm analysis in C++](#). ISBN: 0-201-36122-1

Sedgewick, Robert; [Algorithms in C++](#). ISBN: 0-201-35088-2

Bibliografia Complementar

Deitel, H. M.; [C++ how to program](#). ISBN: 0-13-185757-6

Stroustrup, Bjarne; [The C++ programming language](#). ISBN: 0-201-88954-4

Koenig, Andrew; [Accelerated C++](#). ISBN: 0-201-70353-X

Cormen, Thomas H.; [Introduction to algorithms](#)

Métodos de ensino e atividades de aprendizagem

As aulas teóricas são usadas para exposição formal da matéria, acompanhada da apresentação de exemplos e sua discussão.

Nas aulas práticas são resolvidos exercícios de programação em C++, sendo os exercícios focados em temas tratados nas aulas teóricas. As resoluções são geralmente efetuadas em grupos de estudantes, sendo fomentada a discussão sobre as propostas de solução que vão sendo apresentadas.

A avaliação é realizada ao longo do semestre, em datas previamente anunciadas, onde os estudantes são continuamente avaliados a nível teórico e prático, a nível individual e de grupo. Ao longo do semestre são realizados três pontos de avaliação individual, que incluem questões de índole teórica e realização de exercícios individuais em computador. Ao longo do semestre são também propostos dois projetos de programação a realizar em grupo, fomentando assim a capacidade de trabalho em equipa. O desenvolvimento destes projetos é feito essencialmente fora das aulas, com apanhamento regular.

Software

Google Tests: unit testing

Doxygen: Documentation system for C++

CLion: a cross-platform IDE for C++

Palavras Chave

Ciências Físicas > Matemática > Algoritmos

Ciências Físicas > Ciência de computadores > Programação

Tipo de avaliação

Avaliação distribuída sem exame final

Componentes de Avaliação

Designação	Peso (%)
Teste	60,00
Trabalho laboratorial	40,00
Total:	100,00

Componentes de Ocupação

Designação	Tempo (Horas)
Elaboração de projeto	60,00

Estudo autónomo	40,00
Frequência das aulas	52,00
Trabalho laboratorial	8,00
Total:	160,00

Obtenção de frequência

O estudante não pode exceder o limite de faltas previsto.

Fórmula de cálculo da classificação final

A avaliação será baseada nas seguintes componentes:

CIP = componente prática de programação, a realizar individualmente (média dos momentos de avaliação a realizar durante o semestre)

CIT = componente teórica através de questionários de respostas múltiplas, a realizar individualmente

CG = componente de grupo. Inclui 2 trabalhos (CG1 e CG2) a realizar em grupo (de 3 estudantes), de igual peso, sendo em cada um destes pontos de avaliação contabilizado o empenho e assiduidade do estudante no trabalho de grupo.

Classificação final = 30% CIP + 30% CIT + 40% CG

Observações:

1. É exigida classificação mínima de 40% em todas as componentes CIP, CIT, CG1 e CG2
2. Terá lugar uma prova de recuperação (componente prática + componente teórica) para estudantes admitidos e que não obtiveram a nota mínima nas componentes CIP e/ou CIT. A nota desta prova substitui as avaliações CIP e CIT e é limitada a 50% em cada uma destas.

Provas e trabalhos especiais

A avaliação em Época Especial inclui duas componentes (CE e CT):

- Componente de Exame (CE) (60%), inclui:
 - uma parte prática de programação usando testes unitários (CEP)
 - uma parte teórica (CET)
- Componente de Trabalho Prático(CT) (40%), relativo à implementação de um mini-projeto e elaboração do respetivo relatório. O Trabalho Prático deverá ser entregue até à data do exame e os estudantes devem contactar o regente da UC para escolha do trabalho a implementar.

A classificação final (CF) é computada como: $CF = CE \cdot 0,6 + CT \cdot 0,4$.

Para aprovação na disciplina, o estudante deve obter uma classificação mínima de 40% em qualquer das componentes CEP, CET e CT.

Avaliação especial (TE, DA, ...)

Os estudantes inscritos ao abrigo de regimes especiais sem frequência de aulas práticas:

- podem efetuar a Componente de Grupo (CG) de avaliação constituindo grupo com outros colegas ou sozinhos, devendo acordar com os docentes o trabalho a realizar e as datas de avaliação dos mesmos
- devem realizar a avaliação individual CIP e CIT nas datas estipuladas.

Melhoria de classificação

A realizar na época seguinte da unidade curricular