

Universidade de Aveiro

Programação II

2º semestre 2015/2016

1 Dados gerais

UC n.º:	43257
Semestre:	2.º
Escolaridade semanal:	2h (teórico-práticas) + 3h (práticas)
Unidades de crédito:	8 ECTS
Audiência:	Mestrado Integrado em Engenharia Electrónica e Telecomunicações (MIEET) Mestrado Integrado em Engenharia de Computadores e Telemática (MIECT) Licenciatura em Matemática (LM)
Área/sub-área científica:	Informática / Ciência e Tecnologia da Programação

2 Corpo docente

Coordenação:	Luís Seabra Lopes – ls1@ua.pt
Aulas teórico-práticas:	António J.S. Teixeira – ajst@ua.pt João Manuel Rodrigues – jmr@ua.pt Luís Seabra Lopes – ls1@ua.pt Óscar Pereira – omp@ua.pt
Aulas práticas:	António J.S. Teixeira – ajst@ua.pt Carlos Bastos – cbastos@ua.pt Flávio Meneses – flaviosmeneses@ua.pt João Manuel Rodrigues – jmr@ua.pt José Moreira – jose.moreira@ua.pt Luís Seabra Lopes – ls1@ua.pt Óscar Pereira – omp@ua.pt Osvaldo Pacheco – orp@ua.pt Susana Mota – smota@ua.pt Tomás Oliveira e Silva – tos@ua.pt

3 Objectivos

No final desta unidade curricular os estudantes deverão estar habilitados a:

- Identificar e construir algoritmos iterativos e recursivos de baixa e média complexidade.
- Construir programas usando uma decomposição modular das soluções e recorrendo a tipos de dados abstractos.
- Desenvolver e utilizar diferentes estruturas de dados (pilha, fila, listas, árvores, tabelas de dispersão) bem como dos algoritmos associados para pesquisa, introdução e remoção de informação.

4 Pré-requisitos

Noções elementares de programação usando o paradigma imperativo. Conhecimentos elementares de Java.

5 Programa

Revisões e Introdução ao Linux na perspectiva do utilizador

Linguagem de comandos bash

Suporte para programação em Java

Revisões (Programação I, incluindo manipulação de ficheiros de texto)

Estrutura dos programas em Java

Classes

Pacotes

Aproximações modulares à programação

Avaliação da qualidade de programas

Modularidade

Evolução na modularidade na construção de programas

Protecção de informação (encapsulamento)

Tipos de dados abstractos

Correcção e robustez

Mecanismo de Excepções

Aproximações sistemáticas à programação

Gestão de Falhas em Módulos

Introdução à Programação por Contrato

Recursividade

Abordagem *top-down* vs. *bottom-up* à implementação de processos iterativos

Backtracking

Ordenação e complexidade algorítmica

Ordenação por inserção, fusão e *quicksort*

Introdução à complexidade algorítmica

Estruturas de dados
 Pilhas, Filas
 Tipos de dados genéricos
 Listas ligadas e bi-ligadas
 Memórias Associativas e Tabelas de dispersão
 Árvores binárias de procura
Abordagens sistemáticas à resolução de problemas

6 Bibliografia

- *Estruturas de Dados e Algoritmos em Java*, António Adrego da Rocha, FCA, Abril 2011
- *Programação, Algoritmos e Estruturas de Dados*, João Pedro Neto, Escolar Editora, 2004
- *Java – How to Program*, Deitel H.M. e Deitel P.J., Pearson Education International / Prentice Hall, 6th Edition, 2005
- *Problem Solving with Java*, Koffman E.B. e Wolz U., Addison Wesley, 2nd Revised Edition, 2002
- *Data Structures and Algorithms in Java (4th edition)*, Michael T. Goodrich and Roberto Tamassia, John Wiley & Sons, Inc.
- *Data Structures & Algorithms in Java*, Mitchell Waite Signature Series,
- Documentação on-line de apoio ao ambiente de desenvolvimento de programas em Java fornecida pela Sun Microsystems Inc (Java 2 JDK Standard Edition 6)

7 Modelo de Ensino/Aprendizagem

A unidade curricular está organizada em blocos de aulas teórico-práticas e práticas. Cada um desses blocos decorre entre as 14h e as 19h de um determinado dia da semana (segunda-, terça- ou quinta-feira). Cada aluno frequenta turmas teórico-práticas e práticas de um mesmo bloco.

O objectivo das aulas teórico-práticas é a apresentação e análise de um tópico específico da matéria da unidade curricular. A abordagem seguida pressupõe quase sempre a participação dos estudantes na discussão, procurando-se desenvolver competências que visam o raciocínio abstracto, a manipulação simbólica e a aprendizagem de técnicas gerais de resolução de problemas.

A componente prática é composta por um conjunto de trabalhos que deverão ser sempre finalizados com sucesso, de forma a que os estudantes consolidem os conceitos teórico-práticos discutidos.

Para além das aulas existirão sessões de Orientação Tutorial (abertas a todos), que servirão para discussão e clarificação de dúvidas. Horário: segundas, terças e quintas, das 19 às 20h (a presença de um docente deverá ser garantida através de contacto prévio com o coordenador, ls1@ua.pt)

8 Balanceamento do calendário letivo

Dado existir no calendário escolar da Universidade de Aveiro para o presente semestre (e especificamente para os dias em que são leccionadas aulas de Programação II) uma discrepância de 13 para 15 semanas letivas para diferentes dias da semana, são implementadas as seguintes alterações: ¹

- No dia 06/04/2015, quarta-feira, a partir das 15h, é leccionada uma aula suplementar a cada uma das turmas TP e P de segunda-feira
- No dia 26/04/2015, terça-feira, não são leccionadas aulas de Programação II

Desta forma, todas as turmas terão exactamente 14 semanas de aulas.

9 Regras de Avaliação

Esta unidade curricular segue o modelo de avaliação discreta, com três momentos de avaliação ao longo do período lectivo complementados com uma avaliação final:

	Data de realização	Peso
(API_1) Avaliação Prática Intercalar n.º 1	16 de Março	20%
(API_2) Avaliação Prática Intercalar n.º 2	4 de Maio	20%
($ATPI$) Avaliação Teórico-Prática Intercalar	1 de Junho	20%
(APF) Avaliação Prática Final	Época normal de exames	40%

A classificação global é determinada pela média das notas obtidas com os pesos indicados acima.

As avaliações práticas (API_1 , API_2 e APF) estão centradas no desenvolvimento de programas de acordo com especificações dadas. A avaliação teórico-prática ($ATPI$) envolve a resposta a perguntas sobre a matéria teórica, partindo por vezes da análise de programas dados.

A primeira avaliação (API_1) será realizada em grupos de dois alunos. As restantes avaliações serão realizadas de forma individual. Todas as avaliações são realizadas usando os computadores dos laboratórios de informática do Complexo Pedagógico e do Departamento de Electrónica, Telecomunicações e Informática.

9.1 Exame da época de recurso

O exame da época de recurso consistirá das seguintes avaliações:

- APR - avaliação prática similar às anteriores (API_1 , API_2 e APF), abrangendo toda a matéria prática; e
- $ATPR$ - avaliação teórico-prática similar a $ATPI$, abrangendo toda a matéria teórico-prática.

Cada aluno poderá fazer as duas avaliações, ou apenas uma delas. Quem fizer as duas terá a nota global calculada como $0.8 \cdot APR + 0.2 \cdot ATPR$. Quem fizer apenas APR , terá a nota global calculada como $0.8 \cdot APR + 0.2 \cdot ATPI$. Quem fizer apenas $ATPR$, terá a nota global calculada como $0.2 \cdot ATPR + 0.2 \cdot API_1 + 0.2 \cdot API_2 + 0.4 \cdot APF$.

¹Esta alteração está aprovada pelo Conselho Pedagógico.

9.2 Faltas

Serão registadas as presenças em todas as aulas teórico-práticas e práticas. Para os alunos ordinários (sem estatuto de trabalhador-estudante), o acesso aos exames e a obtenção de aprovação na disciplina ficam condicionados ao registo de um número mínimo de presenças de acordo com o *Regulamento de Estudos*.

As justificações de eventuais faltas devem ser entregues na secretaria do Departamento de Electrónica, Telecomunicações e Informática. Convém notar que, de acordo com as regras em vigor na Universidade de Aveiro, a justificação de faltas, independentemente da razão subjacente, terá de ser feita num prazo máximo de 10 dias após a ocorrência.

