

PROGRAMA DA UNIDADE CURRICULAR

Índice

Objectivos da aprendizagem	2
1. Compreender, saber implementar e manipular estruturas de dados do tipo <i>array</i> ;	2
2. Compreender, saber implementar e manipular pilhas e filas de espera;.....	2
3. Compreender e saber implementar técnicas de programação recursivas;	2
4. Compreender, saber implementar e manipular <i>linked lists</i> (listas encadeadas);.....	2
5. Compreender, saber implementar e manipular árvores.	2
Programa.....	2
1. Introdução à Algoritmia e Estruturas de Dados.....	2
2. Vectores – Arrays	2
3. Pilhas e Filas de Espera	2
4. Recursividade – Conceitos e técnicas.....	2
5. Listas encadeadas.....	3
6. Árvores.....	3
Metodologia de ensino e avaliação	3
Demonstração de coerência entre metodologias de ensino e resultados de aprendizagem.....	3
Bibliografia	4
Fundamental:.....	4
Complementar:	4
Internet:	4
Notas do professor.....	5
Pré-requisitos.....	5
Ferramenta para desenvolvimento	5

Objectivos da aprendizagem

Para concluir com sucesso esta unidade curricular, os alunos deverão demonstrar possuir os seguintes conhecimentos e capacidades:

1. Compreender, saber implementar e manipular estruturas de dados do tipo *array*;
2. Compreender, saber implementar e manipular pilhas e filas de espera;
3. Compreender e saber implementar técnicas de programação recursivas;
4. Compreender, saber implementar e manipular *linked lists* (listas encadeadas);
5. Compreender, saber implementar e manipular árvores.

Os algoritmos e estruturas de dados serão implementados usando as linguagens C e C++.

Programa

1. Introdução à Algoritmia e Estruturas de Dados

2. Vectores – Arrays

- 2.1. Introdução aos vectores
- 2.2. Implementação de um vector
- 2.3. Adicionar, alterar e remover elementos de um vector
- 2.4. Ordenação de um vector
- 2.5. Pesquisa sequencial e pesquisa binária ou dicotómica

3. Pilhas e Filas de Espera

- 3.1. O que são Pilhas
 - 3.1.1. Implementar uma Pilha
 - 3.1.2. Adicionar, alterar e remover elementos a uma Pilha
- 3.2. O que são Filas de Espera
 - 3.2.1. Implementar uma Fila de Espera
 - 3.2.2. Adicionar, alterar e remover itens a uma fila de espera

4. Recursividade – Conceitos e técnicas

- 4.1. Recursividade – Conceitos base
- 4.2. Função Factorial
- 4.3. Recursão excessiva – *Fibonacci Numbers*
- 4.4. *Tail Recursion*
- 4.5. Problemas na utilização de funções recursivas

5. Listas encadeadas

5.1. Listas simplesmente encadeadas

5.1.1. Criar uma lista simplesmente encadeada

5.1.2. Percurso e localização de um item

5.1.3. Adicionar, alterar e remover itens em qualquer ponto de uma lista simplesmente encadeada

5.2. Listas duplamente encadeadas

5.2.1. Implementar uma lista duplamente encadeada

5.2.2. Adicionar, alterar e remover itens de uma lista duplamente encadeada

5.3. Listas encadeadas circulares

5.3.1. Implementar uma lista encadeada circular

5.3.2. Adicionar, alterar ou remover um conjunto de itens a uma lista encadeada circular

6. Árvores

6.1. Conceitos base

6.2. Representação de árvores

6.3. Árvores binárias

6.4. Árvores de ordenação binária

6.5. Atravessamento de uma árvore

6.6. Localizar e inserir itens em árvores de ordenação binária

6.7. Implementar uma árvore de ordenação binária

6.8. Inserir um nó numa árvore de ordenação binária

6.9. Verificar se existe um dado nó numa árvore de ordenação binária

6.10. Calcular o número de nós da árvore de ordenação binária

Metodologia de ensino e avaliação

As aulas desta unidade curricular são de natureza teórico-prática. Estão previstas 60 horas de contacto. O tempo total de trabalho do aluno corresponde a 162 horas. Em todas as aulas, exercícios de aplicação prática dos algoritmos e das estruturas de dados complementam a exposição teórica dos apresentados. Esta metodologia permite que os alunos adquiram, não apenas os conhecimentos teóricos, mas também as necessárias competências para aplicar as estruturas de dados a situações práticas simuladas.

De acordo com o Regulamento de Funcionamento do ISTECS a avaliação é efectuada através de um exame escrito individual e obrigatório. Na classificação final, poderão ser considerados elementos de avaliação contínua, tais como testes, trabalhos individuais ou em grupo, assim como a participação nas aulas presenciais e em recursos de aprendizagem proporcionados por sistemas de e-learning.

Demonstração de coerência entre metodologias de ensino e resultados de aprendizagem

A obtenção dos objectivos da unidade curricular é assegurada pela natureza teórico-prática das aulas da unidade curricular que são planeadas para permitir a compreensão teórica e prática dos conceitos, partindo das estruturas de dados mais simples para as construções mais complexas.

Bibliografia

Fundamental:

Rocha, António A., Estruturas de Dados e Algoritmos em C, 2014, 3ª. edição, FCA, ISBN 978-972-722-769-3.

Shaffer, Clifford A. Data Structures and Algorithm Analysis in C++, Third Edition (2011)

Complementar:

Goodrich, Michael T.; Tamassia, Roberto e Mount, David M. Data Structures and Algorithms in C++, Second Edition, John Wiley & Sons

Internet:

Acesso a publicações da especialidade, gratuitamente, através da rede SPRINGER: <https://link.springer.com/>

Notas do professor

Pré-requisitos

Para compreender o conteúdo desta unidade, o aluno deve estar na posse dos conhecimentos transmitidos na unidade curricular **Programação I**, particularmente os conceitos e manipulação de:

- Arrays
- Strings
- Structures
- Pointers
- Objectos
- Funções (passagem de parâmetros)
 - Valor
 - Referência de endereço
 - Pointer
- Recursividade

Também deve ser capaz de:

- Gerar um executável a partir do código
- Utilizar o debugger
- Conhecer um ambiente integrado de desenvolvimento (IDE) para C++

Ferramenta para desenvolvimento

Nesta unidade utilizar-se-á um IDE que pode ser:

- DevC++
 - Bom para aprender, pois “ajuda” pouco com sugestões
 - O debugger é pouco evoluído
- Visual Studio (VS)
 - É suficiente a versão gratuita, Community
 - Ambiente muito complexo
 - Gera um enorme conjunto de pastas e ficheiros
- Visual Studio Code (VSCode)
 - Equilibrado entre “ajuda” e complexidade
 - Precisa de um compilador, que não faz parte da ferramenta
 - Tem a vantagem de funcionar em várias plataformas (Windows, Linux e Mac)
 - É extremamente versátil devido às extensões que se podem adicionar
 - **Este é o que recomendo**

Documentação auxiliar

Documento para ajuda à instalação do Visual Studio Code, bem como do compilador de C++.

Documento com os passos para criação de uma aplicação usando o Visual Studio.