

### **TeSP - Desenvolvimento de Jogos Digitais**

Técnico Superior Profissional

Plano: Aviso de Registo nº R/Cr 55/2017 de 13-07-2017

#### **Ficha da Unidade Curricular: Fundamentos de Algoritmos**

ECTS: 5; Horas - Totais: 135.0, Contacto e Tipologia, TP:60.0;

Ano|Semestre: 1|S1; Ramo: Tronco comum;

Tipo: Obrigatória; Interação: Presencial; Código: 63903

Área de educação e formação: Ciências informáticas

#### **Docente Responsável**

Fernando Sérgio Hortas Rodrigues

Professor Adjunto

#### **Docente e horas de contacto**

Jorge Miguel Oliveira Lourenco Ferreira

Assistente Convidado, TP: 60;

#### **Objetivos de Aprendizagem**

Aprender a modelar, em linguagem sequencial e estruturada, soluções de uso quotidiano, que poderão ser necessárias implementar numa máquina digital, em ambiente de desenvolvimento de linguagem C++.

#### **Objetivos de Aprendizagem (detalhado)**

Os alunos terão que aprender a focar-se num determinado problema (ou parte) proposto para programar e, com recurso a algoritmos já estudados ou a implementar, chegar a uma conclusão lógica e racional, de forma que a solução encontrada tenha um mínimo de custo de tempo e ciclos de execução.

O facto de o aluno ter noção de que os algoritmos, já existentes, podem ajudá-lo na solução de um determinado problema, em parte ou num todo, poderá criar nele uma metodologia de alcance rápido ao desenvolvimento.

O objectivo centra-se essencialmente na abstracção e enfoque, sabendo que um problema complexo pode ser dividido em pequenas soluções, das quais, depois de agregadas numa sequência lógica, levarão à solução da questão inicial. O aluno terá que, no final, conseguir partir de um objectivo global e dele desenvolver as etapas, com recurso a ferramentas/algoritmos já implementados/conhecidos ou com recurso aqueles que possa haver necessidade de criar, para ter o resultado criado em software funcional.

#### **Conteúdos Programáticos**

1. Arquitectura computacional básica
2. Algoritmos e linguagem "alto nível"
3. Paradigmas da programação
4. Representação de um algoritmo num Fluxograma
5. Ambiente Integrado de Desenvolvimento
6. Linguagem C++
7. Fluxo de controlo de execução
8. Entrada e Saída de dados
9. Declarações e Definições de tipos básicos de dados em C++
10. Enumerados e Structs
11. Arrays
12. Definição e Declaração de funções em C++
13. Estudo aprofundado de alguns algoritmos

## **Conteúdos Programáticos (detalhado)**

1. Arquitectura computacional básica
  - 1.1 – Representação simples de um microcomputador
    - 1.1.1 – Memória, CPU, INPUT/OUTPUT
    - 1.1.2 – Tipos básicos de dados máquina ( bit, byte )
  2. Algoritmos e linguagem “alto nível”
    - 2.1 – A execução de software numa máquina – definição de “baixo nível”
    - 2.2 – Simplificação da programação e desenvolvimento de software em “alto nível”
  3. Paradigmas da programação
    - 3.1 – Programação procedimental sequencial
    - 3.2 – Programação modular e sua representação
  4. Representação de um algoritmo num Fluxograma
    - 4.1 – Introdução aos gráficos de um fluxograma
      - 4.1.1 – Representação de uma acção
      - 4.1.2 – Definição de variável, representação de afectação
      - 4.1.3 – Representação de uma comparação
      - 4.1.4 – Representação de entrada e saída de valores
    - 4.2 – Ligação entre grafos e sentido de execução
    - 4.3 – Execução, Input e Output
    - 4.4 – Chamada a subfunções
  5. Ambiente Integrado de Desenvolvimento
    - 5.1 – Breve introdução ao VisualStudio 2008
  6. Liguagem C++
    - 6.1 – Conversão entre a liguagem visual, fluxograma, e equivalentes primitivos do C++
    - 6.2 – Definição de tipos básicos de variáveis ( char, int, float ) e noção de signed/unsigned
    - 6.3 – Definição do tipo booleano e tratamento de “verdadeiro” (diferente de zero) /”falso” (igual a zero)
      - 6.3.1 – Comparação vs Afectação ( ==, !=, <, <=, =>, >, = )
      - 6.3.2 - Implicações na utilização de valores de variáveis numa comparação ou acção booleana
  7. Fluxo de controlo de execução
    - 7.1 – Iniciação aos diversos tipos de controlo de fluxo na linguagem C++
      - 7.1.1 – ciclos Do/While – Representação do fluxo de execução com fluxograma
      - 7.1.2 – ciclos While – Representação do fluxo de execução com fluxograma
      - 7.1.2 – ciclos For – Representação do fluxo de execução com fluxograma
    - 7.2 – Inicialização dos tipos de comparação de dados básicos na liguagem C++
      - 7.2.1 – Comparação directa if / else
      - 7.2.2 – Multipla comparação switch / case
    - 7.3 – Quebras e repetição de ciclos – como implementar as execpções ilustradas nos fluxogramas
      - 7.3.1 – primitivas break e continue
  8. Entrada e Saída de dados
    - 8.1 – Como implementar a saída e entrada de dados de um fluxograma para o equivalente C++
    - 8.2 – Livrarias de tratamento de I/O na consola (conio.h stdio.h)
  9. Declarações e Definições de tipos básicos de dados em C++
    - 9.1 – Como definir e afectar valores para os tipos básicos de dados
      - 9.1.1 – Limites de valores possíveis
      - 9.1.2 – Noção de overflow e underflow
      - 9.1.3 – Noção de variáveis globais e locais de uma função
  10. Enumerados e Structs
    - 10.1 – Criação de estruturas personalizáveis de dados. Necessidade e sua aplicação
    - 10.2 – Definição de uma estrutura de dados, em grafo
      - 10.2 – Definição no C++ do equivalente grafo
        - 10.2.1 – Acesso e afectação dos elementos da estrutura.
  11. Arrays
    - 11.1 – Definição de array e sua necessidade numa estrutura computacional
    - 11.2 – Limites práticos de utilização e memória disponível

- 11.3 – Representação grafo de um array e acesso leitura/escrita
- 11.4 – Implementação equivalente no C++ de um array e respectivo acesso
- 12. Definição e Declaração de funções em C++
- 12.1 – Desenvolvimento de funções em C++
- 12.1.1 – Parametros de entrada
- 12.1.2 – Retorno de valor
- 12.1.3 – Tipo void para funções que apenas efectuam trabalho e nada retornam
- 12.2 – Do fluxograma, subrotina, ao C++ equivalente
- 12.2.1 – A função inicial de um fluxograma no equivalente C++ - a função main()
- 12.3 – Desenvolver grafos de algoritmos tendo em conta a limitação da linguagem C++
- 12.3.1 – Funções Iterativas
- 12.3.2 – Funções Recursivas
- 12.4 – Manipulação prática de dados e arrays
- 12.4.1 – Ordenação de dados – Algoritmo bubblesort e quicksort
- 12.4.2 – Definição e manipulação de String – Concatenação, ordenação
- 12.4.3 – Definição de matrizes (arrays) multi-dimensionais – visualização grafo e acesso aos elementos
- 12.4.3.1 – Representação de um tabuleiro e equivalente de matriz bi-dimensional
- 12.4.3.2 – Representação de um cubo e equivalente de matriz tri-dimensional
- 12.4.4 – Arrays de estruturas, aplicações práticas
- 12.4.4.1 – Acesso ao elementos
- 12.4.4.2 – O construtor de uma estrutura para inicialização dos elementos básicos em determinado valor
- 12.5 – Tabelas de busca

### **Metodologias de avaliação**

- a) Avaliação contínua;
- b) Realização de trabalho prático final, individual ou em grupo de 2 alunos.

Cada componente terá que ter nota mínima de 10 valores;  
A nota final terá um peso de 30% de a) e 70% de b)

### **Software utilizado em aula**

Poderão haver duas vertentes, do ambiente de desenvolvimento, dado que o objectivo é o desenvolvimento de aplicações em consola:

Linux: Code::blocks IDE, gcc  
Windows: VisualStudio 2008.

### **Estágio**

NA

### **Bibliografia recomendada**

Programação em C++ Conceitos básicos e Algoritmos, Pimenta Rodrigues, Pedro Pereira, Manuela Sousa.  
ISBN 972-722-038-X

### **Coerência dos conteúdos programáticos com os objetivos**

São abordados, inicialmente, os conceitos e as ferramentas para tratamento de algoritmos, baseados numa disposição gráfica de fluxo (Fluxogramas). Utiliza-se de seguida uma linguagem de programação procedimental, a linguagem C++, para aplicar os conceitos transmitidos.

### **Metodologias de ensino**

Aulas Teórico-práticas onde é dada a matéria a leccionar e são aplicados pequenos problemas a resolver, de acordo com o curso da mesma.

### Coerência das metodologias de ensino com os objetivos

Os conceitos teóricos são lecionados com recurso a exemplos simples, práticos e ilustrativos. Na componente prática são abordados os mesmos exercícios, de uma forma mais detalhada, para uma consolidação profunda dos conceitos, uma vez que na componente prática é que se podem testar algumas falhas nos algoritmos anteriormente desenvolvidos, levando a uma melhor reflexão num problema posterior.

### Língua de ensino

Português

### Pré requisitos

Conhecimentos de Matemática.

Saber trabalhar com ambiente de computador na edição de texto.

Domínio de análise e resolução de problemas.

### Programas Opcionais recomendados

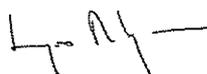
Análise de Matemática no domínio de Matrizes;

Língua estrangeira Inglês;

### Observações

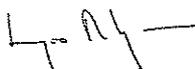
---

#### Docente Responsável



Digitally signed by Fernando Sérgio Hortas Rodrigues  
DN: c=PT, l=Tomar, o=Instituto Politécnico de Tomar, ou=Tecnologias de Informação e Comunicação, cn=Fernando Sérgio Hortas Rodrigues  
Date: 2020.08.26 10:56:09 +01'00'

#### Diretor de Curso, Comissão de Curso



Digitally signed by Fernando Sérgio Hortas Rodrigues  
DN: c=PT, l=Tomar, o=Instituto Politécnico de Tomar, ou=Tecnologias de Informação e Comunicação, cn=Fernando Sérgio Hortas Rodrigues  
Date: 2020.08.26 10:56:28 +01'00'

#### Conselho Técnico-Científico

Jorge  
Antunes

Digitally signed by Jorge Antunes  
DN: cn=Jorge Antunes, o=IPT,  
ou=ISTA,  
email=jorge.antunes@ipt.pt, c=PT  
Adobe Acrobat Reader v6.0.2  
2020.08.26 10:56:28