

**TeSP - Tecnologia e Programação em Sistemas de Informação**

Técnico Superior Profissional

Plano: Despacho n.º 12805/2021 - 29/12/2021

**Ficha da Unidade Curricular: Programação II**

ECTS: 5; Horas - Totais: 135.0, Contacto e Tipologia, TP:55.0;

Ano | Semestre: 1 | A

Tipo: Obrigatória; Interação: Presencial; Código: 602413

Área de educação e formação: Ciências informáticas

**Docente Responsável**

Fernando Sérgio Hortas Rodrigues

Professor Adjunto

**Docente(s)**

Fernando Sérgio Hortas Rodrigues

Professor Adjunto

José Pedro Franco Rodrigues

Assistente Convidado

**Objetivos de Aprendizagem**

Aprender a linguagem de Programação C e aplicar os conceitos básicos da programação de computadores na resolução de problemas utilizando esta linguagem.

**Objetivos de Aprendizagem (detalhado)**

1. Adquirir os conhecimentos sobre os conceitos básicos da linguagem C, tais como: Tipos de dados; estruturas sequenciais; estruturas de decisão; estruturas de repetição.
2. Aprender a implementar o conceito de Modularidade com recurso aos mecanismos disponibilizados pela linguagem C
3. Aprender o conceito de recursividade e como implementá-lo utilizando a linguagem C
4. Adquirir conhecimentos sobre aspetos avançados da linguagem de programação C, tais como: apontadores; funções e passagem de parâmetros; estruturas; memória dinâmica.
5. Aprender a utilizar os mecanismos de persistência de dados existentes na linguagem C.

**Conteúdos Programáticos**

1. Introdução à Linguagem C;
2. Estruturas de Decisão Condicional;
3. Estruturas de Repetição;
4. Tipos de Dados Compostos (arrays; strings);
5. Modularidade (Procedimentos e Funções);
6. Recursividade;
7. Apontadores;
8. Gestão Dinâmica de Memória;
9. Estruturas;
10. Ficheiros;

### **Conteúdos Programáticos (detalhado)**

1. Introdução à Linguagem C
  - 1.2. Tipos de dados básicos
  - 1.3. Variáveis e constantes
  - 1.4. Leitura de dados
  - 1.5. Escrita de informação
  - 1.6. Operadores
    - 1.6.1. Sobre inteiros
    - 1.6.2. Sobre reais
    - 1.6.3. Sobre caracteres
  - 1.7. Casting
2. Estruturas de Decisão Condicional
  - 2.1. Valores lógicos
  - 2.2. Operadores lógicos e relacionais
  - 2.3. Expressões lógicas
  - 2.4. Decisão condicional
    - 2.4.1. Instrução if-else
    - 2.4.2. Ciclos encadeados
  - 2.5. Decisão por escolha
    - 2.5.1. Instrução switch
    - 2.5.2. Instrução break
  - 2.6. Operador de decisão :?
  - 2.7. Blocos de instruções
    - 2.7.1. Indentação
3. Estruturas de Repetição
  - 3.1. Repetição com controlo inicial
    - 3.1.1. Ciclo while
    - 3.1.2. Ciclo for
  - 3.2. Repetição com controlo final
    - 3.2.1. Ciclo do-while
  - 3.3. Alteração do fluxo de execução do ciclo
    - 3.3.1. Instrução break
    - 3.3.2. Instrução continue

- 3.4. Ciclos encadeados
- 3.5. Ciclos infinitos
- 3.6. Operadores de incremento e decrementos
- 3.7. Atribuição composta

#### 4. Tipos de Dados Compostos (arrays; strings)

- 4.1. Arrays
  - 4.1.1. Noção de array
  - 4.1.2. Declaração e inicialização de arrays
  - 4.1.3. Leitura escrita e processamento de arrays
  - 4.1.4. Arrays multidimensionais
  - 4.1.5. Passagem de arrays para funções
- 4.2. Strings
  - 4.2.1. Strings e caracteres
  - 4.2.2. Declaração e inicialização de strings
  - 4.2.3. Leitura escrita e processamento de strings
  - 4.2.4. Passagem de strings para funções
  - 4.2.5. Biblioteca string.h

#### 5. Modularidade (Procedimentos e Funções)

- 5.1. Abordagem top-down
- 5.2. Características das funções
- 5.3. Parâmetros de uma função
- 5.4. Instrução return
- 5.5. O tipo "void"
- 5.6. Variáveis globais e locais
- 5.7. Bibliotecas de Funções

#### 6. Recursividade

#### 7. Apontadores

- 7.1. Conceitos básicos
- 7.2. Declaração e inicialização de apontadores
- 7.3. Ponteiros e tipos de dados
- 7.4. Ponteiros e arrays
- 7.5. Aritmética de ponteiros
- 7.6. Ponteiros de ponteiros
- 7.7. Passagem de parâmetros
  - 7.7.1 Passagem por valor
  - 7.7.2 Passagem por referência
  - 7.7.4 Os parâmetros argc e argv da função main

#### 8. Gestão Dinâmica de Memória

- 8.1. Alocação dinâmica de memória
- 8.1. Libertação de memória
- 8.2. Utilização de memória dinâmica

#### 9. Estruturas

- 9.1. Conceito de estrutura
- 9.2. Declaração de estruturas
- 9.3. Declaração de variáveis do tipo estrutura
- 9.4. Definição de tipos: typedef
- 9.5. Estruturas aninhadas
- 9.10. Arrays de estruturas
- 9.11. Ponteiros para estruturas
- 9.12. Passagem de estruturas para funções

## 10. Ficheiros

- 10.1. Ficheiros vs Streams
- 10.2. Abertura de um ficheiro
- 10.3. Modos de abertura
  - 10.3.1. Modo de texto
  - 10.3.2. Modo binário
- 10.4. Leitura e escrita de caracteres num ficheiro
- 10.5. Fecho de um ficheiro
- 10.6. Input e Output formatado
- 10.7. Ficheiros standard (stdin, stdout, stderr, stderr, stderr)
- 10.8. Processamento de ficheiros binários
  - 10.8.1. Escrita de blocos em ficheiros binários
  - 10.8.2. Leitura de blocos em ficheiros binários
- 10.9. EOF (End of File)
- 10.10. Acesso sequencial e direto a ficheiros
- 10.11. Posicionamento específico num ficheiro

## **Metodologias de avaliação**

Todas as épocas de avaliação

- Prova escrita - 40% com um mínimo de 7 valores em 20;
- Trabalho Prático - 60% - com um mínimo de 10 valores em 20.

Observação 1: A nota da prova escrita obtida em época de frequência pode ser aproveitada para época normal de exame, se o aluno não tiver obtido aprovação na época de frequência.

Observação 2: O Trabalho prático têm defesa obrigatória. A falta à defesa implica atribuição de zero (0) valores nessa componente e a consequente reprovação na época de avaliação em causa.

## **Software utilizado em aula**

C Language Compiler - GCC  
WSL - Windows Subsystem for Linux

## **Estágio**

Não Aplicável.

### **Bibliografia recomendada**

- Kernighan, B. e Ritchie, D. (1988). *C Programming Language* (pp. 1-270). 2º, Prentice Hall. EUA  
- Damas, L. (1999). *Linguagem C* (pp. 1-628). 1ª, FCA. Lisboa

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

Os capítulos 1, 2, 3 e 4 permitem atingir o objetivo 1 e é também suporte para atingir grande parte dos restantes objetivos. O capítulo 5 permite atingir o objetivo 5. As matérias constantes do capítulo 6 contribuem diretamente para o capítulo 6. Os capítulos 7, 8 e 9 permitem atingir o objetivo 4. O capítulo 10 permite atingir o objetivo 5.

### **Metodologias de ensino**

Aulas expositivas para apresentação dos conteúdos teóricos do programa; Aulas práticas laboratoriais para resolução de problemas e consolidação de conhecimentos utilizando o computador; Apoio tutorial para esclarecimento de dúvidas.

### **Coerência das metodologias de ensino com os objetivos**

Os conceitos teóricos subjacentes aos vários objetivos são lecionados com recurso a técnicas expositivas e a exemplos simples e ilustrativos de aplicação. Na componente prática laboratorial os alunos efetuam exercícios e/ou tutoriais com exemplos mais complexos e completos, para uma consolidação profunda dos conceitos transmitidos.

### **Língua de ensino**

Português

### **Pré-requisitos**

Não Aplicável.

### **Programas Opcionais recomendados**

Não Aplicável.

### **Observações**

Objetivos de Desenvolvimento Sustentável:

- 4 - Garantir o acesso à educação inclusiva, de qualidade e equitativa, e promover oportunidades de aprendizagem ao longo da vida para todos;
  - 5 - Alcançar a igualdade de género e empoderar todas as mulheres e raparigas;
  - 9 - Construir infraestruturas resilientes, promover a industrialização inclusiva e sustentável e fomentar a inovação;
  - 12 - Garantir padrões de consumo e de produção sustentáveis;
- 

**Docente responsável**

---