

**TeSP - Manutenção de Sistemas Mecatrónicos**

Técnico Superior Profissional

Plano: Despacho n.º 11230/2020 - 13/11/2020

**Ficha da Unidade Curricular: Programação**

ECTS: 5; Horas - Totais: 135.0, Contacto e Tipologia, TP:30.0; PL:45.0; OT:2.50;

Ano | Semestre: 1 | S1

Tipo: Obrigatória; Interação: Presencial; Código: 61213

Área de educação e formação: Ciências informáticas

**Docente Responsável**

Bruno Miguel Santana Chaparro

Professor Adjunto

**Docente(s)**

Bruno Miguel Santana Chaparro

Professor Adjunto

**Objetivos de Aprendizagem**

Pretende-se dotar os alunos com os conhecimentos necessários sobre algoritmia e programação, conhecer e utilizar os programas para resolver problemas de engenharia nesse ambiente.

**Objetivos de Aprendizagem (detalhado)**

Tratando-se de uma disciplina de introdução à programação, pretende-se que os alunos compreendam as diferenças entre linguagens de baixo e de alto nível, saibam as vantagens e inconvenientes de ambas, e sejam capazes de selecionar a linguagem mais adequada para a resolução de problemas específicos. Os conteúdos da disciplina têm também como objetivo preparar os alunos com o nível de conhecimentos adequado sobre as linguagens apresentadas para que possam no futuro desenvolver de forma autónoma os seus projetos nesta área.

**Conteúdos Programáticos**

1. Algoritmos.
2. Programação em Linguagem de Alto Nível.

- 3. Programação Matemática
- 4. Bases de Dados
- 5. Programação de Microprocessadores.

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

- 1. Algoritmos.
  - 1.1 Definições básicas.
  - 1.2 A Lógica no desenvolvimento de programas.
  - 1.3 Fluxogramas.
  - 1.4 Estruturas de controlo.
- 2. Programação em linguagem de alto nível.
  - 2.1 Introdução.
    - 2.1.1 Programas.
    - 2.1.2 Compiladores.
    - 2.1.3 Ambientes de programação.
    - 2.1.4 Variáveis.
    - 2.1.5 Tipos de dados.
    - 2.1.6 Inicialização de variáveis
    - 2.1.7 Conversão entre diferentes tipos de dados
  - 2.2 Estrutura de Programação.
    - 2.2.1 Expressões.
    - 2.2.2 Instruções.
    - 2.2.3 Estruturas condicionais.
    - 2.2.4 Funções.
    - 2.2.5 Vetores.
  - 2.3 Erros.
    - 2.3.1 Tipos de erros.
    - 2.3.2 Debugging.
  - 2.4 Técnicas de programação.
    - 2.4.1 Estrutura de um programa.
    - 2.4.2 Classes.
    - 2.4.3 Entrada e saída de dados.
- 3. Programação Matemática
  - 3.1 Introdução.
    - 3.1.1 Variáveis.
    - 3.1.2 Matrizes.
    - 3.1.3 Operadores.
  - 3.2 Estrutura de Programação
    - 3.2.1 Funções.
    - 3.2.2 Estruturas condicionais.
    - 3.2.3 Ciclos for e while.
    - 3.2.4 Scripts.
    - 3.2.5 Debugging.
    - 3.2.6 Importação e exportação de dados.
  - 3.3 Ferramentas de visualização.
    - 3.3.1 Gráficos xy.

- 3.3.2 Gráficos 3D.
- 3.4 Cálculo simbólico.
  - 3.4.1 Declaração de variáveis simbólicas.
  - 3.4.2 Resolução analítica de equações.
- 5. Programação de Microprocessadores
- 4. Introdução às Bases de Dados
  - 4.1. Estruturas de Bases de Dados
  - 4.2. Tipos de Variáveis
  - 4.3. Implementação de Bases de Dados
- 5.1. Introdução
- 5.2. Entradas e Saídas
- 5.3. Tipos de interface com o utilizador
- 5.4. Programação de placas de desenvolvimento

### **Metodologias de avaliação**

Época de avaliação por frequência:

- trabalhos teórico-práticos (50%)
- prova teórico-prática (50%)

Restantes épocas:

- prova teórico-prática (100%)

### **Software utilizado em aula**

Visual Studio  
MariaDB  
Octave  
Arduino IDE

### **Estágio**

Não aplicável.

### **Bibliografia recomendada**

- Lopes, A. (2010). *Introdução à Programação em Visual Basic 2010*. 2ª Edição, FCA. Lisboa

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

A aprendizagem teórico-prática e o trabalho laboratorial disponibilizam conhecimentos sobre técnicas de programação e desenvolvimento de sistemas informáticos. De forma a capacitar a análise de um problema, estimular a compreensão das técnicas básicas de linguagens de programação, em problemas reais.

### **Metodologias de ensino**

Nas aulas TP são transmitidos conhecimentos sobre as linguagens de programação, acompanhadas de análise e discussão de algoritmos. Nas aulas PL, os estudantes são orientados no treino de técnicas programação.

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

As metodologias de aprendizagem assentam no método expositivo dos conteúdos programáticos definidos e em trabalhos práticos. O trabalho prático permitirá uma aproximação à aprendizagem prática em ambiente industrial.

### **Língua de ensino**

Português

### **Pré-requisitos**

Não aplicável.

### **Programas Opcionais recomendados**

Não aplicável.

### **Observações**

---

### **Docente responsável**

---