

**TeSP - Informática**

Técnico Superior Profissional

Plano: Despacho n.º 8838/2020 de 14-09-2020 + Despacho n.º 3463/2023 de 16/03/2023

**Ficha da Unidade Curricular: Programação Avançada**

ECTS: 6; Horas - Totais: 162.0, Contacto e Tipologia, TP:70.0;

Ano | Semestre: 1 | S2

Tipo: Obrigatória; Interação: Presencial; Código: 61428

Área de educação e formação: Ciências informáticas

**Docente Responsável**

Fernando Sérgio Hortas Rodrigues

Professor Adjunto

**Docente(s)**

Fernando Sérgio Hortas Rodrigues

Professor Adjunto

Fernando Jorge Lima dos Santos Barros

Assistente Convidado

**Objetivos de Aprendizagem**

Esta unidade curricular tem por objectivo introduzir os alunos às linguagens de programação. Ao concluir esta unidade o estudante deverá saber estruturar e escrever programas com recurso a linguagens de programação de alto nível, nomeadamente através da linguagem de programação Python.

**Objetivos de Aprendizagem (detalhado)**

Esta unidade curricular tem por objectivo introduzir os alunos à programação de computadores de alto nível, nomeadamente através da linguagem de programação Python. Ao concluir esta unidade o estudante deverá:

- 1) ter conhecimento profundo das características desta linguagem de programação e saber instalar e configurar o ambiente de desenvolvimento;
- 2) ter conhecimento dos principais comandos da linguagem de programação

- 3) ter conhecimento das principais bibliotecas existentes em Python;
- 4) saber automatizar rotinas com recurso a estruturas de controlo e iteração;
- 5) saber escrever e estruturar programas em Python com recurso a arrays;
- 6) saber escrever e estruturar programas em Python com recurso a estruturas de dados avançadas;
- 7) saber escrever e estruturar programas em Python com recurso a compreensão de listas;
- 8) ser capaz de processar ficheiros de texto, word, CSV, JSON, e documentos web, bem como entender a importância do OCR (mecanismo de reconhecimento de caracteres) no processamento de texto a partir de imagens e pdfs;
- 9) saber decompor problemas em sub-tarefas com recurso a funções reutilizáveis e anónimas;
- 10) saber criar e disponibilizar módulos;
- 11) ser capaz de executar módulos/scripts a partir da linha de comandos;
- 12) estar capacitado para testar e corrigir erros em programas;
- 13) ser capaz de gerir logs decorrentes de exceções.

### **Conteúdos Programáticos**

1. Programação em Python
2. Introdução ao Python
3. Importação e Utilização de Bibliotecas
4. Estruturas de Controlo e Iteração
5. Estruturas de Dados Simples
6. Estruturas de Dados Avançadas
7. Compreensão de Listas
8. Leitura e Escrita de Ficheiros
9. Funções
10. Criação e Partilha de Módulos
11. Linha de Comandos
12. Exceções
13. Logs
14. Introdução à Orientação a Objetos

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

1. Programação em Python
  - 1.1. Porquê programar em Python?
  - 1.2. História do Python
  - 1.3. Características
  - 1.4. Vantagens
  - 1.5. Instalação do Python
2. Introdução ao Python
  - 2.1. Comentários
  - 2.2. Ajuda no Python
  - 2.3. Entrada e saída de informação
  - 2.4. Variáveis
  - 2.5. Tipos de dados

- 2.6. Operadores
- 2.7. Casting
- 2.8. Formatação de dados
- 2.9. Imutabilidade vs Mutabilidade
- 2.10. Notebooks em Python
  
- 3. Importação e Utilização de Bibliotecas
  - 3.1. Módulos internos
  - 3.2. Módulos externos
  - 3.3. Módulos frequentemente usados
  - 3.4. Introdução ao PyPi: repositório oficial de pacotes do Python
  - 3.5. Criação de Ambiente Virtuais
  
- 4. Estruturas de Controlo e Iteração
  - 4.1. IF
  - 4.2. For
  - 4.3. While
  - 4.4. Break/Continue
  
- 5. Estruturas de Dados Simples
  - 5.1. Arrays
  - 5.2. Arrays Multidimensionais
  - 5.3. Jagged Arrays
  
- 6. Estruturas de Dados Avançadas
  - 6.1. Listas
  - 6.2. Conjuntos
  - 6.3. Dicionários
  - 6.4. Tuples
  - 6.5. Names Tuples
  - 6.6. Enums
  
- 7. Compreensão de Listas e LINQ
  - 7.1. Introdução à compreensão de listas
  - 7.2. LINQ em Python
  
- 8. Leitura e Escrita de Ficheiros
  - 8.1. Ficheiros de texto
  - 8.2. Ficheiros de Imagem
  - 8.3. Ficheiros PDF
  - 8.4. Ficheiros MS Word
  - 8.5. Ficheiros HTML
  - 8.6. Ficheiros CSV
  - 8.7. Ficheiros JSON
  
- 9. Funções
  - 9.1 Funções definidas pelo utilizador
  - 9.2. Funções geradoras

### 9.3 Funções lambda (MAP, Filter, Reduce)

## 10. Criação e Partilha de Módulos

### 10.1 Introdução ao git (sistema de controlo de versões)

### 10.2 Introdução ao github (repositório de código-fonte com ligação ao git)

### 10.3 Sincronização de projetos com o github, a partir do Git e do software de desenvolvimento

## PyCharm

### 10.4. Criação de módulos

### 10.5. Disponibilização local de módulos

### 10.6. Disponibilização online de módulos (PyPi e Github)

### 10.7. Criação de pacotes

### 10.8. Disponibilização local de pacotes

### 10.9. Disponibilização online de pacotes (PyPi e Github)

## 11. Linha de Comandos

### 11.1. Execução de código a partir da linha de comandos

### 11.2. Passagem de parâmetros

### 11.3. Módulo `__name__`

### 11.4. Função Main

## 12. Exceções

### 12.1. Definição

### 12.2. Tipos de erros

### 12.3. Controlo de exceções

## 13. Logs

### 13.1 Níveis de mensagens de logs

### 13.2 Formatação de logs

### 13.3 Mensagens de Logs

### 13.4 Controlo de exceções no âmbito de logs

### 13.5 Escrita de logs em ficheiros

## 14. Introdução à Orientação a Objetos

### 14.1 Classes

### 14.2 Instancias de Classe

### 14.3 Construtores

### 14.4 Atributos de classe estáticos

### 14.5 Métodos

### 14.6 Iteradores

## **Metodologias de avaliação**

Avaliação por Frequência:

Frequência I (40%) [Prova com consulta, Tudo menos IAs]

Frequência II (60%) [Com Cons. Parcial: Apenas folhas com notas escritas à mão].

Os alunos deverão ter, em cada um dos elementos de avaliação, uma nota mínima de 6 valores em 20.

A classificação final da UC resulta da média ponderada das classificações obtidas nas componentes de avaliação definidas.

O aluno obtém aprovação à UC, estando dispensado de Exame, de acordo com o disposto nos Pontos 11 e 12, do Artigo 11º, do regulamento Académico do IPT.

Avaliação por Exame: Exame (100%) [Prova sem consulta].

O aluno obtém aprovação à UC, estando dispensado de Exame, de acordo com o disposto nos Pontos 11 e 12, do Artigo 11º, do regulamento Académico do IPT.

Requisito obrigatório de admissibilidade à frequência e ao exame:

- Entrega de um mínimo de 80% dos problemas de programação propostos nas aulas;

### **Software utilizado em aula**

Python - Anaconda

Jupyter Notebooks

Git

PyCharm

### **Estágio**

Não aplicável.

### **Bibliografia recomendada**

- Carvalho, A. (2021). *Práticas de Python - Algoritmia e Programação*. FCA. Lisboa
- Costa, E. (2024). *Programação em Python: Fundamentos e Resolução de Problemas*. 2, FCA. Lisboa
- Downey, A. (0). *Think Python - How to Think Like a Computer Scientist*. Acedido em 16 de fevereiro de 2018 em <http://greenteapress.com/wp/think-python>
- Severance, C. (0). *Python for Everybody - Exploring Data Using Python 3*. Acedido em 16 de fevereiro de 2018 em [http://do1.dr-chuck.com/pythonlearn/EN\\_us/pythonlearn.pdf](http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)

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

Os conteúdos programáticos estão em coerência com os objetivos da unidade curricular, atendendo a que:

- O ponto 1 dos conteúdos programáticos pretende concretizar o ponto 1 dos objetivos
- O ponto 2 dos conteúdos programáticos pretende concretizar o ponto 2 dos objetivos
- O ponto 3 dos conteúdos programáticos pretende concretizar o ponto 3 dos objetivos
- O ponto 4 dos conteúdos programáticos pretende concretizar o ponto 4 dos objetivos
- O ponto 5 dos conteúdos programáticos pretende concretizar o ponto 5 dos objetivos
- O ponto 6 dos conteúdos programáticos pretende concretizar o ponto 6 dos objetivos
- O ponto 7 dos conteúdos programáticos pretende concretizar o ponto 7 dos objetivos
- O ponto 8 dos conteúdos programáticos pretende concretizar o ponto 8 dos objetivos
- O ponto 9 dos conteúdos programáticos pretende concretizar o ponto 9 dos objetivos
- O ponto 10 dos conteúdos programáticos pretende concretizar o ponto 10 dos objetivos

- O ponto 11 dos conteúdos programáticos pretende concretizar o ponto 11 dos objetivos
- O ponto 12 dos conteúdos programáticos pretende concretizar o ponto 12 dos objetivos
- O ponto 13 dos conteúdos programáticos pretende concretizar o ponto 13 dos objetivos
- O ponto 14 dos conteúdos programáticos pretende concretizar o ponto 14 dos objetivos

### **Metodologias de ensino**

Aulas teórico-práticas expositivas onde se descrevem os conceitos fundamentais. Aulas práticas-laboratoriais de resolução de casos práticos e aplicação dos conceitos a cenários de utilização real.

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

Os objetivos de aprendizagem do curso são atingidos através da realização de um conjunto de exercícios práticos permitindo desta forma que os alunos solidifiquem as competências adquiridas. Considera-se ainda importante a orientação tutorial, onde o docente procura esclarecer dúvidas e apontar soluções para o sucesso do processo de aprendizagem da UC.

### **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;
- 11 - Tornar as cidades e comunidades inclusivas, seguras, resilientes e sustentáveis;

---

### **Docente responsável**

---