

## LICENCIATURA EM ENGENHARIA INFORMÁTICA

### FUNDAMENTOS DE PROGRAMAÇÃO

*PROPOSTAS DE TRABALHO*



## Trabalho



## Missão:

A missão consiste em escolher (ou propor um tema) de entre as várias propostas seguintes e desenvolver os algoritmos que resolva o problema apresentado.

Deve elaborar um pequeno relatório com o algoritmo desenvolvido especificado em fluxograma (parcial), pseudocódigo (parcial), e codificado na linguagem Python. Apresente um exemplo com a prova e teste do algoritmo (parcial).

1. Red Bull Air Race
2. Se7eMaravilhas
3. Mundial20xx, Europeu 20xx ou outro campeonato (Liga, Europeu, etc.)
4. Mergulho
5. Óscares
6. Presidentes EUA ou de Portugal
7. Gestor Pessoal de Biblioteca ou Músicas
8. Jogo
9. Proposta do estudante

## Avaliação

Indicadores de avaliação	Pontos (0-20)
Algoritmos	10
Funcionalidade	4
Legibilidade	1
Modularidade	2
Relatório + Apresentação	3

### Observações

- Requisitos principais: deve recorrer à modularização com funções e ficheiros de texto (CSV – *Comma Separate Values*)
- O trabalho só é considerado aceite, após defesa do mesmo. A defesa do trabalho é individual, na defesa serão realizadas perguntas sobre o código disponibilizado;
- Deve entregar via Moodle, o relatório e o projeto desenvolvido em Python;
- Pode acrescentar outras opções que valorizem o trabalho
- Data de entrega: Janeiro 2025;
- Data de defesa: a combinar

Esta mensagem vai autodestruir-se em cinco minutos!

**Proposta 1****Red Bull Air Race**

A organização do Red Bull Air Race pretende um programa para analisar estatisticamente o campeonato. Para cada uma das 8 corridas, pretende-se guardar num vector (lista), `vecRaces` o nome da cidade, num vector (lista) `vecPilotos` o piloto que venceu a corrida e num vector (lista) `vecTempos` o tempo obtido. Pode substituir os vetores anteriores com uma lista de listas.

A introdução de dados deve ser processada da seguinte forma: para cada uma das 8 corridas, o utilizador deve introduzir o nome da cidade e depois introduz o nome do vencedor e o tempo obtido. Estes dados devem ser guardados nos respetivos vetores ou outras estruturas como lista de listas.

**Menu:**

1. Introdução dados
2. Geração dados (preenchimento dos vetores (Listas) de forma automática)
3. Alterar dados
4. Eliminar dados
5. Consultar
6. Pesquisar (ver dados de uma determinada corrida)
7. Grelha de classificação
8. Guardar em ficheiro de texto
9. Carregar dados do ficheiro
10. Sair

Deve recorrer à modularização do programa a desenvolver, isto é, procedimentos e funções.

**Obs: pode / deve definir outras opções, bem como usar outros dados adicionais de forma a enriquecer a aplicação.**

## Proposta 2

### Se7eMaravilhas

A comissão organizadora das 7 Maravilhas de Portugal pretende um programa para analisar estatisticamente os resultados de uma das edições das se7emaravilhas, por exemplo, os 21 monumentos, que se realizou a 07/07/2007.

O programa deve pedir para cada um dos 21 monumentos finalistas, a sua votação final. Pretende-se armazenar apenas a votação final num vector (lista).

Cada posição do vector (lista) representa respetivamente, os seguintes monumentos:

- |                                |                                |
|--------------------------------|--------------------------------|
| 1 - Castelo de Almoral         | 2 - Castelo de Guimarães       |
| 3 - Castelo de Marvão          | 4 - Castelo de Óbidos          |
| 5 - Convento de Cristo Tomar   | 6 - Convento de Mafra          |
| 7 - Fortaleza de Sagres        | 8 - Fortificações de Monsaraz  |
| 9 - Igreja São Francisco Porto | 10 - Torre dos Clérigos        |
| 11 - Mosteiro da Batalha       | 12 - Mosteiro de Alcobaça      |
| 13 - Mosteiro dos Jerónimos    | 14 - Paço Ducal de Vila Viçosa |
| 15 - Universidade de Coimbra   | 16 - Palácio de Mateus         |
| 17 - Palácio Nacional da Pena  | 18 - Palácio de Queluz         |
| 19 - Ruínas de Conímbriga      | 20 - Templo Romano de Évora    |
| 21 - Torre de Belém            |                                |

A votação final é obtida da seguinte forma: para cada monumento é pedida a votação a 5 peritos numa escala de 0 a 100. A votação final é a média das 5 votações.

#### Menu:

1. Introdução dados
2. Geração dados (preenchimento do vector (lista) de forma automática)
3. Alterar dados
4. Eliminar dados
5. Consultar
6. Pesquisar (ver dados de um monumento)
7. Monumento vencedor
8. Guardar em ficheiro / Carregar dados do ficheiro
9. Sair

Deve recorrer à modularização do programa a desenvolver, isto é, procedimentos e funções.

**Obs: pode / deve definir outras opções, bem como usar outros dados adicionais de forma a enriquecer a aplicação.**

## Proposta 3



## Mundial20xx, Europeu 20xx

Ou outro campeonato (Liga, Europeu, etc..)

A organização do Mundial20xx precisa de um programa para fazer a análise de espectadores do campeonato do Mundo de 20xx da 1ª Fase. Os dados que se pretende guardar está representada na seguinte tabela (lista de lista).

Jogo	Grupo	Seleção 1	Seleção 2	Estádio	Espectadores	VIPs
1	A	Africa Sul	Mexico	Ellis Park Stadium		
2	A	Uruguai	França	Green Point Stadium		
3	B	Argentina	Nigeria	Ellis Park Stadium		
4	B	Rep. Korea	Grecia	Nelson Mandela Bay Stadium		
..						
30	G	Portugal	Coreia Norte	Green Point Stadium		
...						
48						

OBS: a informação de cada coluna pode ser guardada num vector (lista) ou recorrer a estruturas (lista de listas) ou matrizes.

### Menu:

1. Introdução dados
2. Geração dados (preenchimento dos vectores (lista) ou das matrizes de forma automática)
3. Alterar dados
4. Eliminar dados
5. Consultar
6. Pesquisar (ver dados de um jogo)
7. Totais espectadores
8. Total de espectadores (de uma seleção, em todos os jogos)
9. Guardar em ficheiro
10. Carregar dados do ficheiro
11. Sair

Deve recorrer à modularização do programa a desenvolver, isto é, procedimentos e funções.

**Obs: pode / deve definir outras opções, bem como usar outros dados adicionais de forma a enriquecer a aplicação.**

## Proposta 4

**Mergulho**

A federação olímpica de mergulho, precisa de uma solução para o tratamento das notas obtidas pelos nadadores durante uma competição.

Na competição olímpica de mergulho as notas de um nadador são atribuídas da seguinte forma: oito juízes dão notas entre 0 e 10 valores. Em seguida as notas mais baixa e mais alta são descartadas e a nota final do nadador é obtida através da média das outras 6 (seis) notas restantes.

Para cada um dos 10 nadadores, além das notas obtidas é perguntado o seu número. Pretende-se guardar numa matriz ou vectores (listas) todos estes dados. Pode substituir os vetores anteriores com uma estrutura (lista de listas) e um vetor (lista).

Nadador	Nota Juiz 1	Nota Juiz 2	Nota Juiz 3	Nota Juiz 4	Nota Juiz 5	Nota Juiz 6	Nota Juiz 7	Nota Juiz 8	Nota Final
...									
....									

**Menu:**

1. Introdução dados
2. Geração dados (preenchimento da matriz (lista) de forma automática)
3. Alterar dados
4. Eliminar dados
5. Consultar
6. Pesquisar (ver dados de um nadador)
7. Nadador vencedor
8. Guardar em ficheiro
9. Carregar dados do ficheiro
10. Sair

Deve recorrer à modularização do programa a desenvolver, isto é, procedimentos e funções.

**Obs: pode / deve definir outras opções, bem como usar outros dados adicionais de forma a enriquecer a aplicação.**

## Proposta 5

**Oscars**

Suponha que lhe foi solicitado apoio na organização do concurso dos Óscares, na categoria de cinema para a seleção do melhor filme. Assim, pretende-se que desenvolva um algoritmo para registar o nome de 10 filmes e as respetivas pontuações. As pontuações são atribuídas numa escala de 0 a 20 pontos. Estes dados devem ser armazenados em dois vectores (lista), um para o nome dos filmes e outro para as pontuações (sendo a correspondência do nome do filme - pontuação efetuada pelo índice do vector (lista)). Pode substituir os vetores anteriores com uma estrutura e um vetor (lista).

Ordene os vectores (lista) por ordem crescente da pontuação obtida. Apresente os nomes dos filmes e respetivas pontuações, colocando em destaque (num pódio, (opção 6)) os três filmes com maior pontuação e os restantes filmes alinhados por baixo (apresente o nome do filme e pontuação).

**Menu:**

1. Introdução dados
2. Geração dados (preenchimento dos vetores (lista) de forma automática)
3. Alterar dados
4. Eliminar dados
5. Consultar
6. Pesquisar (ver dados de um filme)
7. Pódio
8. Guardar em ficheiro
9. Carregar dados do ficheiro
10. Sair

Deve recorrer à modularização do programa a desenvolver, isto é, procedimentos e funções.

**Obs: pode / deve definir outras opções, bem como usar outros dados adicionais de forma a enriquecer a aplicação.**



## Proposta 6

**Presidentes EUA ou de Portugal**

Pretende-se um programa para analisar estatisticamente os mandatos dos presidentes EUA. Para cada um dos 45 presidentes vamos guardar em vectores (lista), o ano de início do mandato, o ano de fim do mandato e o partido.

Sugestão: quatro vectores (lista) – **vecPresidente**, **vecAnoInicio**, **vecAnoFim**, **vecPartido**. Pode substituir os vetores anteriores com uma estrutura (lista de listas).

Exemplo para o vector vecPresidente. Deve pesquisar os outros dados

```
vecPresidente(1) = "George Washington"
vecPresidente(2) = "John Adams"
vecPresidente(3) = "Thomas Jefferson"
vecPresidente(4) = "James Madison"
vecPresidente(5) = "James Monroe"
.....
vecPresidente(40) = "Ronald Wilson Reagan"
vecPresidente(41) = "George Herbert Walker Bush"
vecPresidente(42) = "William Jefferson Clinton (Bill Clinton)"
vecPresidente(43) = "George Walker Bush"
vecPresidente(44) = "Barack Hussein Obama II"
```

**Menu:**

1. Introdução dados
2. Geração dados (preenchimento dos vectores (lista) de forma automática)
3. Alterar dados
4. Eliminar dados
5. Consultar
6. Pesquisar (ver dados de um dado presidente)
7. Media de anos de todos os mandatos
8. N° de mandatos de um partido
9. Partido com mais mandatos
10. Tabela de mandatos por partido
11. Guardar em ficheiro
12. Carregar dados do ficheiro
13. Sair

Deve recorrer à modularização do programa a desenvolver, isto é, procedimentos e funções.

**Obs: pode / deve definir outras opções, bem como usar outros dados adicionais de forma a enriquecer a aplicação.**

**Proposta 7****Gestor Pessoal de Biblioteca ou Musicas**

Pretende-se um programa para guardar informações acerca dos livros que temos e ou/ CDs.

**Menu:**

1. Introdução dados
2. Alterar dados
3. Eliminar dados
4. Consultar
5. Pesquisar
6. Guardar em ficheiro
7. Carregar dados do ficheiro
8. Sair

Deve recorrer à modularização do programa a desenvolver, isto é, procedimentos e funções.

**Obs: pode / deve definir outras opções, bem como usar outros dados adicionais de forma a enriquecer a aplicação.**

Proposta 8

**Jogo**

Proposta 9

**Proposta do estudante**