

# Laboratório 0 - Algoritmos e Estruturas de Dados

Prof. Inês Almeida

28 de Março de 2022

Deve fazer clone do repositório para a máquina de trabalho. Deve manter o repositório GitHub atualizado com o seu trabalho.

## Datas

- Enunciado: 28 de março de 2022;
- Entrega no e-learning: 03 de abril de 2022, 23:59;

## Entrega

O código produzido deverá estar disponível no repositório GitHub gerado pelo GitHub Classroom. Deve sempre existir um branch main, onde a versão final deverá ficar disponível.

## Parte 1 - Calculadora (10 valores)

Implemente uma calculadora com as seguintes funcionalidades:

- Somar (S)
  1. Entrada:  
`S_numero_numero`
  2. Saida com sucesso:  
`0_resultado_é_<resultado>.`
  3. Saida com insucesso  
Apenas foi inserido um número:  
`Não_é_possivel_realizar_a_operação.`

### Exemplo operação S

**Exemplo (Sucesso):**

**Entrada**

S 4 5

**Saida**

O resultado é 9.

**Exemplo (Insucesso):**

**Entrada**

S 3

**Saida**

Não é possível realizar a operação.

- Subtrair (SU)

1. Entrada:

SU numero numero

A operação deve ser realizada pela ordem que os números foram introduzidos.

2. Saida com sucesso:

O resultado é <resultado>.

3. Saida com insucesso

O resultado da operação é um número negativo:

Não é possível realizar a operação.

### Exemplo operação SU

**Exemplo (Sucesso):**

**Entrada**

SU 10 5

**Saida**

O resultado é 5.

**Exemplo (Insucesso):**

**Entrada**

S 5 10

**Saida**

Não é possível realizar a operação.

- Multiplicar (M)

1. Entrada:

M\_numero\_numero

2. Saida com sucesso:

O\_resultado\_é\_<resultado>.

3. Saida com insucesso:

Apenas foi inserido um número:

Não\_é\_possivel\_realizar\_a\_operação.

#### Exemplo operação M

##### Exemplo (Sucesso):

##### Entrada

M\_6\_2

##### Saida

O\_resultado\_é\_12.

##### Exemplo (Insucesso):

##### Entrada

M\_6

##### Saida

Não\_é\_possivel\_realizar\_a\_operação.

- Dividir (D)

1. Entrada:

D\_dividendo\_divisor

A operação deve ser realizada pela ordem que os números foram introduzidos.

2. Saida com sucesso:

O\_resultado\_é\_<resultado>.

3. Saida com insucesso:

O divisor é 0:

Não\_é\_possivel\_realizar\_a\_operação.

#### Exemplo operação D

**Exemplo (Sucesso):**

**Entrada**

D\_4\_2

**Saida**

O\_resultado\_é\_2.

**Exemplo (Insucesso):**

**Entrada**

D\_6\_0

**Saida**

Não\_é\_possivel\_realizar\_a\_operação.

- Factorial (F)

1. Entrada:

F\_numero

2. Saida com sucesso:

O\_resultado\_é\_<resultado>.

#### Exemplo operação F

**Exemplo (Sucesso):**

**Entrada**

F\_3

**Saida**

O\_resultado\_é\_6.

## 0.1 Regras

O programa deve ser implementado com recurso a funções e de acordo com a arquitetura Model View Controller (MVC). Deve criar o diretório task1 e criar os seguintes ficheiros:

- view.py onde são invocadas todas as funções necessárias para a implementação do programa.
- model.py onde são definidas as estruturas de dados que representam a informação.
- controller.py onde são implementadas todas as funções necessárias para implementação do programa.
- program.py com main.

## 0.2 Avaliação

Esta parte será avaliada com base em duas componentes: quantitativa (A), e qualitativa (B). A nota final da parte 1 é determinada por  $(0.8 \times A) + (0.2 \times B)$ .

**Avaliação quantitativa:** A avaliação quantitativa será baseada nos testes unitários disponíveis no repositório GitHub deste laboratório.

Instruções	Peso
S	1
SU	1
M	3
D	3
F	2

**Avaliação qualitativa:** A avaliação qualitativa irá considerar que existem várias formas de resolver o problema descrito, mas exige-se a:

- Separação entre interface, dados, e lógica da aplicação
- Justificação clara para as variáveis e operações implementadas
- Adequação da escolha de estruturas de dados e algoritmos para a resolução do problema

## Parte 2 - Quadrado Mágico (10 valores)

Implemente um programa que dado uma matriz quadrada ( número de linhas igual ao número de colunas) verifique se esta é um quadrado mágico ou não. Um quadrado mágico é uma matriz quadrada de números inteiros positivos distintos dispostos de tal forma que a soma dos  $n$  números em qualquer linha diagonal, horizontal e vertical é sempre o mesmo número. Exemplo:

8	1	6
3	5	7
4	9	2

- Verificar Quadrado Mágico (VQM)

1. Entrada:

VQM\_numero\_linhas\_numero\_colunas\_numero\_1...numero\_n

2. Saida com sucesso:

(a) A matriz é um quadrado mágico:

É um quadrado mágico.

(b) A matriz não é um quadrado mágico:

Não é um quadrado mágico.

3. Saida com insucesso

O número de linhas da matriz não é igual ao número de colunas.

Matriz inválida.

### 0.3 Regras

O programa deve ser implementado com recurso a funções e de acordo com a arquitetura Model View Controller (MVC). Deve criar o diretório task2 e criar os seguintes ficheiros:

- view.py onde são invocadas todas as funções necessárias para a implementação do programa.
- model.py onde são definidas as estruturas de dados que representam a informação.
- controller.py onde são implementadas todas as funções necessárias para implementação do programa.
- program.py com main.

### 0.4 Avaliação

Esta parte será avaliada com base em duas componentes: quantitativa (A) e qualitativa (B). A nota final da parte 2 é determinada por  $(0.8 \times A) + (0.2 \times B)$ .

**Avaliação quantitativa:** A avaliação quantitativa será baseada nos testes unitários disponíveis no repositório GitHub deste laboratório.

Instruções	Peso
VQM	10

**Avaliação qualitativa:** A avaliação qualitativa irá considerar que existem várias formas de resolver o problema descrito, mas exige-se a:

- Separação entre interface, dados, e lógica da aplicação

- Justificação clara para as variáveis e operações implementadas
- Adequação da escolha de estruturas de dados e algoritmos para a resolução do problema