

# Trabalho Prático 6

## Pilhas e Filas

DWDM/AICD/EIA

1º semestre 2022/2023

Prof.: Carlos Limão

## Introdução

Neste trabalho vamos implementar Pilhas e Filas e usá-las para resolver problemas.

Para o ajudar na implementação serão fornecidos dois ficheiros:

- `fila_alunos.py`
- `stack_alunos.py`

Cada um dos ficheiros contém o esqueleto de duas classes que se espera que escreva na totalidade. O programa incluído em cada um dos módulos testa a implementação que irá realizar.

## Objetivos

Este trabalho tem dois objetivos:

1. Realizar algumas implementações simples de Pilhas e de Listas.
2. Ganhar alguma experiência de utilização destas estruturas de dados e familiarização com as suas aplicações,

## Preparação

Todos os exercícios deste trabalho serão colocados dentro de um único projeto.

### Passo 1 - Criação de um projeto

Comece por criar um novo projeto no PyCharm chamado `Pilhas_e_Filas`.

### Passo 2 - Adicionar ficheiros Python ao projeto

Neste projeto vamos começar por adicionar um ficheiro por cada uma das estruturas de dados a implementar:

- Um ficheiro Python chamado ***fila.py***.
- Um ficheiro Python chamado ***pilha.py***.
- Um ficheiro Python chamado ***profile.py***.

**Note que o ficheiro `profile.py` fornecido com o enunciado pode ser diferente do que já possui!**

O ponto de partida para cada um dos restantes dois ficheiros é também fornecido juntamente com este enunciado. Os ficheiros fornecidos incluem o **esqueleto de duas classes** que servirão de base à duas implementações de cada estrutura de dados que deverá escrever, bem como um **programa para as testar**.

## Exercício 1 - Pilha (Stack)

Neste exercício é-lhe pedido que realize duas implementações de uma Pilha (Stack) usando a estrutura de dados List do Python. Uma deverá usar operações à cabeça da lista (posição índice 0) e a outra deverá usar operações no fim da lista (append e pop).


### Passo 3 - Escrita dos programas

No ficheiro ***pilha.py*** deverá escrever as classes que implementam duas versões de uma Pilha.

Para isso deverá usar como base o ficheiro ***stack\_alunos.py***, fornecido com este enunciado. Esse ficheiro inclui uma descrição de todas as funções (métodos) que deve escrever.

### Passo 4 - Execução do programa


O ficheiro fornecido com este enunciado inclui também um programa de teste que deverá utilizar para testar as suas implementações. Note que pode usar este programa como um guia, passo a passo, do que deve desenvolver.

 Deve informar o professor, caso detete alguns erros no ficheiro que lhe foi fornecido como ponto de partida para a realização dos exercícios deste trabalho.

**Tenha em atenção que o programa de teste fornecido pode não ser suficiente para testar todos os casos.**

**Pergunta 1:** Use o módulo ***profile.py*** para mostrar as diferenças, em termos do tempo de execução das duas implementações que realizou. Inclua no seu relatório os gráficos com os tempos obtidos para cada implementação e explique os resultados obtidos.

**Pergunta 2:** Na submissão do trabalho, deve incluir, num ficheiro com extensão .zip, um diretório **exercício1** contendo todos os módulos Python que escreveu para este exercício.

 Como sempre, deve ter em atenção que todos os ficheiros escritos por si devem incluir um cabeçalho com as características indicadas no enunciado do 1º trabalho prático.

Todas as funções deverão também incluir uma *docstring* que descreva o objetivo da função, os parâmetros de entrada e os valores de saída.

## Exercício 2 - Fila (Queue)

Uma Fila implementa o serviço normalmente oferecido por uma qualquer fila de espera. A admissão à fila faz-se por uma das extremidades e o atendimento faz-se ao primeiro elemento na outra extremidade (disciplina FIFO).

### Passo 5 - Escrita dos programas

No ficheiro ***fila.py*** deverá escrever as classes que implementam duas versões de uma Fila.

Para isso deverá usar como base o ficheiro ***fila\_alunos.py***, fornecido com este enunciado. Esse ficheiro inclui uma descrição de todas as funções (métodos) que deve escrever.

### Passo 6 - Execução do programa

O ficheiro fornecido com este enunciado inclui também um programa de teste que deverá utilizar para testar as suas implementações. Note que pode usar este programa como um guia, passo a passo, do que deve desenvolver.

⚠ Deve informar o professor, caso detete alguns erros no ficheiro que lhe foi fornecido como ponto de partida para a realização dos exercícios deste trabalho.

**Tenha em atenção que o programa de teste fornecido pode não ser suficiente para testar todos os casos.**

**Pergunta 3:** Use o módulo **profile.py** para mostrar as diferenças, em termos do tempo de execução das duas implementações que realizou. Inclua no seu relatório os gráficos com os tempos obtidos para cada implementação e explique os resultados obtidos.

**Pergunta 4:** Na submissão do trabalho, deve incluir, num ficheiro com extensão **.zip**, um diretório **exercício2** contendo todos os módulos Python que escreveu para este exercício.

⚠ Como sempre, deve ter em atenção que todos os ficheiros escritos por si devem incluir um cabeçalho com as características indicadas no enunciado do 1º trabalho prático.

Todas as funções deverão também incluir uma *docstring* que descreva o objetivo da função, os parâmetros de entrada e os valores de saída.

## Exercício 3 - Palíndromas

Um palíndroma é uma string que é igual quer seja lida da esquerda para a direita, quer da direita para a esquerda. Por exemplo: “Amor a Roma” e “O lobo ama o bolo” são ambos palíndromas.

Use uma Pilha e um Stack para realizar uma implementação de uma função chamada **is\_palindrome()**.

⚠ Das classes que desenvolveu, utilize aquelas com melhor desempenho, para cada um dos casos.

A função deve devolver **True** se a string passada como argumento é um palíndroma.

A assinatura da função é a seguinte:

```
def is_palindrome(text):  
    """  
    Devolve True se text é um palíndroma, False caso contrário.  
    """
```

As soluções puramente iterativas para este problema necessitam de uma das seguinte:

- 2n acessos (para inverter a string e depois comparar cada letra).
- Acessos não sequenciais (comparar elementos do início e do fim da string).

Note, no entanto, que se adicionar uma dada sequência de elementos a uma pilha e a uma fila, pode depois retirá-los facilmente destas estruturas de dados por ordem oposta dada a sua própria nature-

za. Explore este facto para implementar a função pedida com exatamente **n** acessos sequenciais a `text`, onde **n** é `len(text)`.

Note que pode ignorar espaços e pontuação e realizar comparações ignorando o “tamanho da letra” (*case insensitive*). As funções associadas a strings `str.isalpha()` e `str.lower()` podem ser úteis.

Inclua alguns testes numa função `main()`.

Exemplos de palíndromas válidos:

- ""
- "a"
- "aa"
- "aaa"
- "aba"
- "abba"
- "A man, a plan, a canal: Panama"
- "Amor a Roma."
- "O lobo ama o bolo"

**Pergunta 5:** Na submissão do trabalho, deve incluir, num ficheiro com extensão `.zip`, um diretório **exercício3** contendo todos os módulos Python que escreveu para este exercício.

## Conclusão do Trabalho

A entrega do trabalho faz-se submetendo, no Moodle, um ficheiro **PDF**, devidamente identificado (aluno, curso, disciplina, trabalho, data), com as **respostas** às perguntas colocadas, eventualmente justificadas com cópias dos ecrãs.

Caso tal seja solicitado, deve submeter também os ficheiros que contêm os eventuais programas que tenham sido pedidos no enunciado. Neste caso os ficheiros devem ter nomes sugestivos, que indiquem claramente a que se referem (p.e.: `algoritmoX.py` ou `exercícioY.py` admitindo que são ficheiros com programas em Python).

### Notas:

1. Não se limite a usar imagens do ecrã inteiro e esperar que o professor encontre nelas a resposta correta ou a justificação para a sua resposta!
2. Use uma imagem **indicando claramente** (sublinhado, círculo, etc.) a **resposta** ou a **justificação** para uma resposta que deu previamente, consoante o caso.
3. Por outro lado, recortes muito pequenos podem não permitir perceber a que contexto se referem. Os recortes devem incluir o contexto suficiente para que se perceba a que se referem e de onde foram retirados.

**Dica:** Pode usar a **Ferramenta de Recorte (Snip & Sketch** em Inglês) para cortar facilmente zonas do ecrã que queira usar nas suas respostas.