

# Trabalho Prático 7

## Tabelas de Hash

DWD/M/AICD/EIA

1º semestre 2022/2023

Prof.: Carlos Limão

## Introdução

Neste trabalho vamos implementar uma tabela de *hash* e usá-la para resolver problemas.

Para o ajudar na implementação serão fornecidos dois ficheiros:

- ***hashtable\_chaining.py***

Contém uma implementação de uma hash table usando Encadeamento (*Chaining*) como método de resolução de colisões. O programa no final deste módulo testa a implementação fornecida.

- ***profile\_hashtable.py***

Para comparação do desempenho das Hash Tables com Listas e Conjuntos no Python.

Pode obter os ficheiros [aqui](#).

## Objetivos

Este trabalho tem dois objetivos:

1. Realizar uma implementação alternativa de uma tabela de *hash* recorrendo à Sondagem Linear (*Linear Probing*) como método de resolução de colisões.
2. Ganhar alguma experiência de utilização deste tipo de estrutura de dados e familiarização com as suas aplicações.

## Preparação

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

### Passo 1 - Criação de um projeto

Comece por criar um novo projeto no PyCharm chamado HashTable.

### Passo 2 - Adicionar ficheiros Python ao projeto

Neste projeto vamos começar por adicionar os dois ficheiros fornecidos:

- Um ficheiro Python chamado ***hashtable\_chaining.py***.
- Um ficheiro Python chamado ***profile\_hashtable.py***.

O ficheiro ***hashtable\_chaining.py*** contém uma implementação funcional de uma Hash Table e usa uma fila em cada posição da tabela de hash como estratégia para a resolução de colisões. Inclui também um **programa de teste**.

O ficheiro ***profile\_hashtable.py*** mede o desempenho da pesquisa de nomes únicos (sem duplicados) numa lista telefónica.

Para isso recorre à *package* **faker** para gerar uma lista telefónica aleatória e em seguida constroi a lista de nomes únicos com um **Dictionary**, um **Set** e uma implementação da **HashTable** existente em *hashtable\_chaining.py*.

## Exercício 1 - Hash Table

Neste exercício é-lhe pedido que realize uma implementação alternativa de uma tabela de Hash que utilize a estratégia **Linear Probing** (sondagem linear) como mecanismo de resolução de colisões.

Recorde que na aula teórica vimos o pseudo-código dos algoritmos das funções `get()`, `put()` e `remove()` a usar neste caso.

### Passo 3 - Adicionar ficheiro Python ao projeto

Deve incluir no projeto um novo ficheiro Python chamado *hashtable\_probing.py* para conter a implementação alternativa que irá realizar.


### Passo 4 - Escrita dos programas

Como ponto de partida para o ficheiro *hashtable\_probing.py* pode usar o conteúdo do ficheiro *hashtable\_chaining.py*.

No novo ficheiro deverá escrever a classe que implementa a tabela de hash recorrendo aos algoritmos acima indicados. Para isso terá fundamentalmente de modificar os métodos `put()`, `get()` e `delete()` da classe `HashTable`. Note que deve manter o nome da classe!


### Passo 5 - Execução do programa

O ficheiro *hashtable\_chaining.py* fornecido com este enunciado inclui também um programa de teste que deverá também utilizar para testar a sua nova implementação.

 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:** Certifique-se que a sua classe funciona corretamente (todos os testes passam sem erros) antes de passar ao exercício seguinte. Certifique-se também que o método `put()` gera uma exceção se a tabela de *hash* estiver cheia.

 Tenha em atenção que quando remove uma entrada da tabela de *hash* e está a usar Linear Probing, não deve marcar a entrada como `None`! Se o fizer, não poderá recuperar o tratamento que fez das colisões nessa posição.

Uma solução comum é usar um valor especial (`VAZIO`) que indica que “estava aqui um elemento, mas foi removido”. Quando fizer um `get()` procede como de costume, mas tendo em atenção que “`VAZIO`” **não conta** como `None`, e que deve prosseguir com a pesquisa. Para recuperar as posições vazias, basta que as considere iguais a `None` na função `put()`.

**Pergunta 2:** Use o módulo *profile\_hashtable.py* para mostrar as diferenças, em termos do tempo de execução das duas implementações. Note que este módulo contém já o código necessário para testar a sua nova implementação. Bastará apenas descomentá-lo. Inclua no seu relatório os dados com os tempos obtidos e explique os resultados obtidos.

**Pergunta 3:** 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.

## 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.