

**NOTA:** Esta ficha está dividida em duas partes (Parte I e Parte II), pretende-se que a primeira parte seja seguida com os slides da aula teórico-prática. Na segunda parte pretende-se que o aluno consiga realizar os exercícios pondo em prática a matéria abordada nos slides e praticada na Parte I. Devem ser consultados os slides: 2019.ED.Aula08.pdf para a realização desta ficha prática.

## Parte I

### Exercício 1

Criar uma classe que represente uma entidade, por exemplo um carro, assim como algumas que ache relevantes para o representar. Armazenar instâncias dessa entidade num array e usar os seguintes métodos de pesquisa: pesquisa linear, pesquisa binária para encontrar um determinado elemento.

### Exercício 2

Criar uma classe que represente uma entidade, por exemplo um carro, assim como algumas que ache relevantes para o representar. Armazenar instâncias dessa entidade numa lista ligada e usar os seguintes métodos de pesquisa: pesquisa linear, pesquisa binária para encontrar um determinado elemento.

Nota: Os métodos usados no exercício anterior e que se encontram nos slides respectivos à aula teórica terão de ser adaptados para lista ligada.

### Exercício 3

Criar uma classe que represente uma entidade do mundo real e que permita ser comparável para testar os vários métodos de ordenação apresentados na aula teórica. Para poder usar os métodos apresentados nos slides da aula teórica terá de adicionar as várias instâncias da classe criada num array. Testar os vários algoritmos de ordenação apresentados:

- *Selection Sort*
- *Insertion Sort*
- *Bubble Sort*
- *Quick Sort*
- *Merge Sort*

#### Exercício 4

Usar o cenário definido no exercício anterior, mas alterar a estrutura de dados que os armazena. No exercício anterior estávamos a usar um array e neste exercício pretendemos usar uma lista ligada. Como todos os métodos de ordenação estão implementados para serem usados com um *array*, estes terão de ser novamente implementados. Testar e implementar os vários algoritmos de ordenação para uma lista ligada:

- *Selection Sort*
- *Insertion Sort*
- *Bubble Sort*
- *Quick Sort*
- *Merge Sort*

## Parte II

#### Exercício 1

Qual é o algoritmo de pesquisa mais eficiente, a pesquisa binária ou a pesquisa linear?

#### Exercício 2

Qual é o algoritmo de ordenação mais eficiente, o *QuickSort* ou o *Merge Sort*?