

**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: **2020.ED.Aula12.pdf** para a realização desta ficha prática.

## **Parte I**

### **Exercício 1**

Implementar um **grafo não pesado** com recurso a uma **matriz de adjacências**. O grafo deverá permitir realizar travessias tanto em largura como em profundidade.

### **Exercício 2**

Implementar um **grafo não pesado** com recurso a uma **lista de adjacências**. O grafo deverá permitir realizar travessias tanto em largura como em profundidade.

### **Exercício 3**

Implementar um método para determinar o caminho mais curto entre dois nós fornecidos (início e destino). Este método deverá ser adicionado aos grafos implementados nos exercícios anteriores (1 e 2).

### **Exercício 4**

Implementar um **grafo pesado** com recurso a uma **matriz de adjacências**. O grafo deverá permitir realizar travessias tanto em largura como em profundidade.

### **Exercício 5**

Implementar um **grafo pesado** com recurso a uma **lista de adjacências**. O grafo deverá permitir realizar travessias tanto em largura como em profundidade.

### **Exercício 6**

Implementar um método para determinar o caminho mais curto entre dois nós fornecidos (início e destino). Este método deverá ser adicionado aos grafos implementados nos exercícios anteriores (4 e 5).

## Parte II

### Exercício 1

Estudar e implementar o algoritmo de *Dijkstra* para o cálculo do caminho mais curto.

### Exercício 2

Implementar um método para devolver a árvore geradora mínima de um grafo. Aplicar este método à implementação de grafo que entender, seja de lista de adjacências ou de matriz de adjacências.