

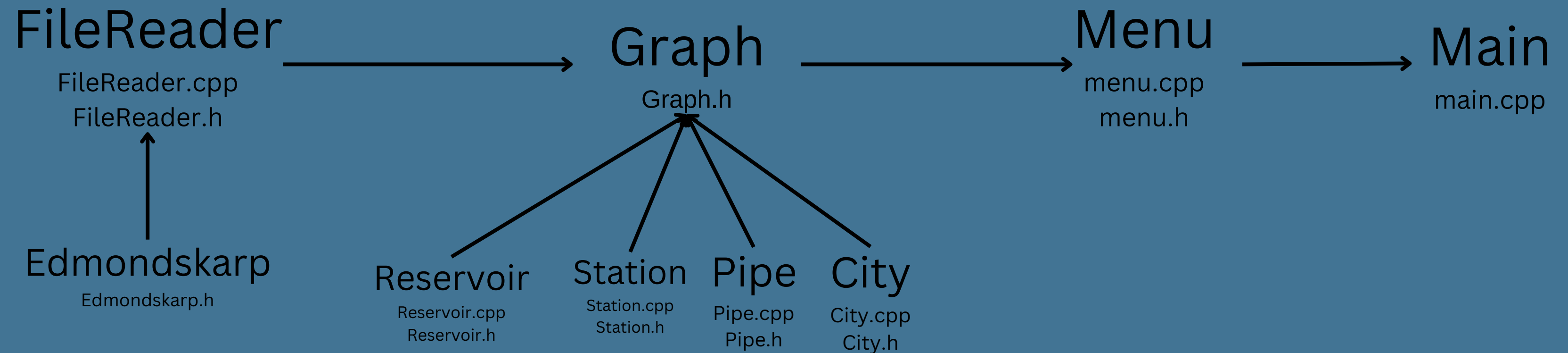
# WATER SUPPLY MANAGEMENT

---

**2LEIC04**

**100% MIGUEL MATEUS->202206944;**

# CLASS DIAGRAM



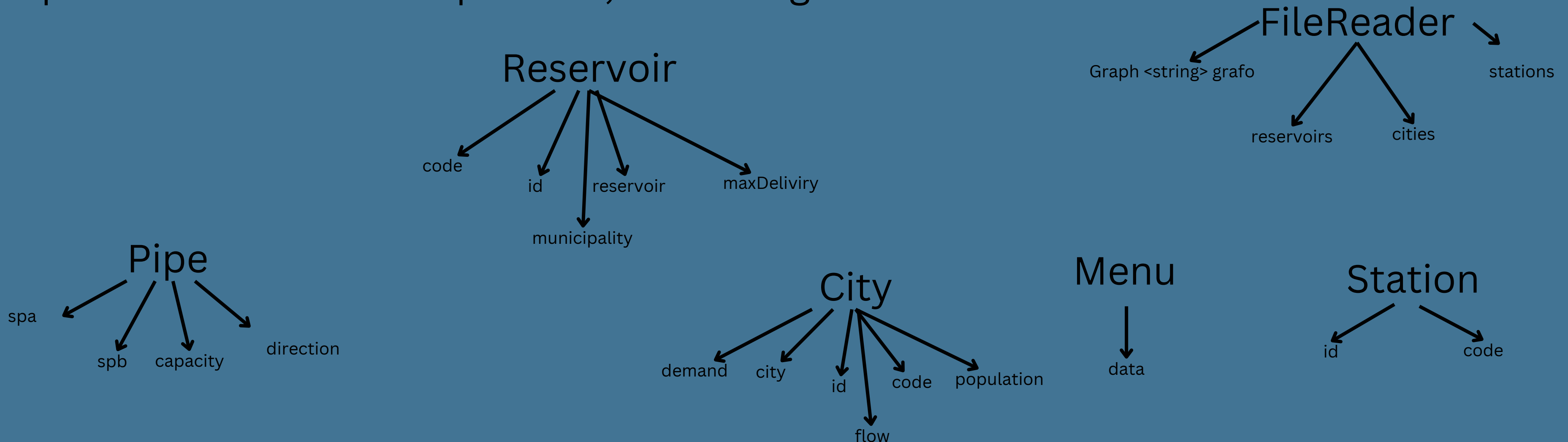
# READING THE GIVEN DATA

Li os ficheiros csv fornecidos para um grafo e também para as hashtables (unordered\_map).

O meu grafo era composto por Vertices<string> e Edges<string> e por isso cada um deles tinha um vetor de nodes e de edges adjacentes, tendo sido essas estruturas de dados algumas das que preenchemos.

# GRAPHS USED TO REPRESENT THE DATASET

O meu Trabalho tem 5 classes criadas por mim, a classe graph.h que é proveniente das aulas práticas e um ficheiro Edmondskarp.h também proveniente das aulas práticas, mas foi ligeiramente alterado.





# GRAPHS USED TO REPRESENT THE DATASET(CONT.)

O meu Grafo é composto por `Vertex<string>` e `Edge<string>`. Eu usei vértices(code de “reservoir”, “city” ou de “station”) e edges(“pipes”) para que seja possível aceder as informações recolhidas dos ficheiros em `Project1DataSetSmall` ou `Project1LargeDataSet`.

Com todas estas informações e com as classes desenhadas na página anterior consegui implementar quase todas as funções do trabalho.

# FUNCIONALIDADES

## SScity

Esta Funcionalidade serve para verificar o “flow” para a cidade para a qual se introduziu um código.

Complexidade:  $O(1)$

## All\_Cities

Esta Funcionalidade serve para verificar o “flow” para todas as cidades individualmente.

Complexidade:  $O(n)$

# FUNCIONALIDADES (CONT.)

## Total

Esta Funcionalidade serve para verificar o “flow” total para toda a “network”

Complexidade:  $O(n)$

## VerifyFlow

Esta Funcionalidade serve para verificar o “flow” de cada cidade individualmente e se este for menor que a “demand” dessa cidade ele apresenta o “decifit”.

Complexidade:  $O(n)$

# FUNCIONALIDADES (CONT.)

## Reservoir\_Failure

Esta Funcionalidade serve para verificar o “flow” de cada cidade individualmente, após a retirada de uma “reservoir” e se este for menor que o “flow” anterior dessa cidade ele apresenta o “decifit”.

Complexidade:  $O(V * E^2)$

## Station\_Failure

Esta Funcionalidade serve para verificar o “flow” de cada cidade individualmente, após a retirada de uma “station” e se este for menor que o “flow” anterior dessa cidade ele apresenta o “decifit”.

Complexidade:  $O(V * E^2)$



# FUNCIONALIDADES (CONT.)

## Pipe\_Failure

Esta Funcionalidade serve para verificar o “flow” de cada cidade individualmente, após a retirada de uma “pipe” e se este for menor que o “flow” anterior dessa cidade ele apresenta o “decifit”.

Complexidade:  $O(V * E^2)$

# User Interface

Menu Principal → Max Flow

```
|      MainMenu      |
| Choose one of the options |
|-----|
| 1. Find Max Water Flow  |
| 2. Verify Water Supply  |
| 3. Balance the network  |
| 4. Reservoir Failure Impact |
| 5. Station Failure Impact |
| 6. Pipeline Failure Impact |
| q. QUIT                 |
|-----|
```

```
| Choose one of the options |
|-----|
| 1. Select Specific City  |
| 2. All Cities            |
| 3. Total                 |
| q. Go back              |
|-----|
Your option:
```

# USER INTERFACE (CONT.)

Esta é um interface simples de usuário baseada no terminal. A navegação neste menu é feita através da escrita do número ou letra que está à esquerda de um título sugestivo á funcionalidade dele.

Por exemplo se no menu principal premir-se 2 ele verifica se todas as “cities” recebem “flow” suficiente para cobrir a sua “demand”.

# FAVORITE FEATURES

Estou, especialmente orgulhoso da funcionalidade do maxflow das cidades individuais, pois foi a primeira a ser implementada e ainda estava ter um bocado de dificuldades com a operação do grafo. O que levou a demorar para entender, que podia, simplesmente, adicionar um getter do flow na classe city, facilitando assim todo o processo.

# DIFFICULTIES

Tive algumas dificuldades em decidir como fazer o grafo, mas acabei por optar pela que me pareceu a opção mais simples. Para além disso, a função para retirar um pipe deu-me dificuldades.

Alguns exercícios podiam ser interpretados de diferentes formas, sendo que por vezes demorávamos a entender por completo o que era pedido. Apesar que com o ficheiro de testes de exemplo enviado ficou mais fácil compreender.



**OBRIGADO**