

# Relatório do Projeto de ESINF

## **Turma\_2DE**

1170500 \_ Hugo Frias

1180730 \_ Vera Pinto

## **Professor**

Nuno Malheiro - NFM

## **Unidade Curricular**

ESINF

---

**Exercício 1a:**

Neste exercício fizemos um load dos dados dos ficheiros txt para um BST em que cada node é um pair de country, número de fronteiras e a lista de fronteiras. Este método de load tem como parâmetros o nome do ficheiro de países e fronteiras para ser mais fácil testar vários ficheiros. O método que recebe o nome de um país e devolve a lista de fronteiras chama-se listOfBorders.

**Exercício 1b:**

Neste exercício fizemos o método OrderedCountries, que recebe um continente por parâmetro e vai à árvore original e passa todos os países pertencentes a esse continente para uma árvore secundária. Depois passa a árvore secundária para uma ordered list usando o inOrder que organiza primeiro decrescente por número de fronteiras e depois crescente por população.

**Exercício 2a:**

Neste exercício carregamos uma árvore 2d-tree através da árvore do 1a. Em que para cada node da BST, compara nos níveis pares a latitude e nos níveis ímpares a longitude.

**Exercício 2b:**

Neste exercício fizemos o método findCountryByCoordinates em que através de duas coordenadas o método percorre a 2d-tree toda e encontra o país com essas exatas coordenadas.

**Exercício 2c:**

No exercício 2C foi-nos pedido para encontrar o vizinho mais próximo de um certo ponto. Para isso usamos o método findNearest. Primeiramente, nós pegamos nas coordenadas do ponto passado por parâmetro e criamos um país imaginário com essas coordenadas. De seguida, vamos "percorrer" a árvore como se fossemos inserir esse tal país, e vamos adicionando a uma lista todos os países percorridos. Dentro desta lista, muito provavelmente vai estar o país mais próximo ao tal ponto. De seguida, calculamos a distância entre todos os países da lista ao ponto imaginário, e vamos guardar a distância mais curta. De seguida iremos utilizar o método desenvolvido no exercício 2D, e vamos buscar a lista de todos os países que estão contidos numa área com um "raio de ação" igual à distância mais curta anterior. De seguida, vamos novamente calcular qual dos países dessa lista é que tem a distância mais curta ao ponto de origem e por fim retornaremos esse país.

**Exercício 2d:**

No exercício 2D foi-nos pedido para devolver uma lista que contenha todos os países que estejam dentro de um "retângulo" delimitado por umas certas coordenadas (latitude, longitude) passadas por parâmetro. Para isso usamos o método getCountriesByArea. Este método vai primeiro verificar a depth em que o país atual se encontra (de modo a saber se vai verificar a latitude ou altitude), caso se encontre numa depth par, ele vai ver se o país tem uma latitude menor que a latitude máxima (caso não tenha, ele vai buscar o LeftNode) e de seguida vai verificar se o país tem uma latitude maior que a latitude

### Diagrama de Classes:

