

O presente projeto envolve a criação de uma biblioteca de classes respetivos métodos e testes relativos a uma **rede social simplificada** que gere uma **rede de amigos** bem como a **rede de cidades** onde habitam e as distâncias entre as mesmas. A informação encontra-se nos ficheiros de texto:

```
users.txt: user,age,city
relationships.txt: user1,user2
countries.txt: country,continent,population,capital,latitude,longitude
borders.txt: country1,country2
```

Nesta rede social **dois utilizadores são amigos** se tiverem uma **ligação direta** entre eles. As **relações de amizade** entre os vários utilizadores devem ser implementadas através de um grafo usando a representação **matriz de adjacências**. O **grafo das cidades** inclui apenas as capitais de países e respetivas distâncias geográficas e deve ser implementado usando a representação **map de adjacências**. Pretende-se que implementem as seguintes funcionalidades:

1. Construir os grafos a partir da informação fornecida nos ficheiros de texto. A capital de um país tem ligação direta com as capitais dos países com os quais faz fronteira. O cálculo da distância em Kms entre duas capitais deverá ser feito através das suas coordenadas¹.
2. Devolver os **amigos comuns** entre os **n utilizadores mais populares** da rede. A popularidade de um utilizador é dada pelo seu número de amizades.
3. Verificar se a rede de amizades é conectada e em caso positivo devolver o número mínimo de ligações necessário para nesta rede qualquer utilizador conseguir contactar um qualquer outro utilizador.
4. Devolver para um utilizador os **amigos que se encontrem nas proximidades**, isto é, amigos que habitem em cidades que distam um dado **número de fronteiras** da cidade desse utilizador. Devolver para cada cidade os respetivos amigos.
5. Devolver as **n cidades com maior centralidade** ou seja, as cidades que em média estão mais próximas de todas as outras cidades e onde habitem pelo menos **p%** dos utilizadores da rede de amizades, onde **p%** é a percentagem relativa de utilizadores em cada cidade.
6. Devolver o **caminho terrestre mais curto** entre dois utilizadores, passando obrigatoriamente pelas **n cidade(s) intermédias** onde cada utilizador tenha o **maior número de amigos**. Note que as cidades origem, destino e intermédias devem ser todas distintas. O caminho encontrado deve indicar as cidades incluídas e a respetiva distância em km.

¹ <http://www.movable-type.co.uk/scripts/latlong.html>

Normas

- A avaliação do trabalho será feita principalmente em função das classes propostas, nomeadamente em termos da sua conformidade com o Paradigma da Programação por Objetos e **eficiência** das estruturas de dados usadas nas funcionalidades solicitadas.
- O projeto tem de ser desenvolvido em Java e todas as funcionalidades testadas através de testes unitários usando os ficheiros de teste disponibilizados, versão small e big.
- O trabalho deverá ser realizado em **grupos de dois alunos**. É obrigatório o uso de uma ferramenta de **controle de versões**.
- O relatório deverá servir de ferramenta de avaliação posterior à apresentação. Nele devem apresentar as classes definidas, **análise de complexidade** de todas as funcionalidades implementadas e melhoramentos possíveis.
- O trabalho deve ser submetido no Moodle até às **24 horas do dia 29 de Novembro**. A partir desta data a nota do trabalho será penalizada **10% por cada dia de atraso** e não se aceitam trabalhos **após dois dias** das datas indicadas.