

# Exercícios POO

## 2 Desenvolvimento da aplicação *Rede Social*

### 2.1 Descrição do problema

O objectivo deste exercício é o desenvolvimento de uma aplicação parecida com o famoso Facebook. A aplicação deve gerir um conjunto de pessoas e um conjunto de relações de amizade entre pessoas. Cada *pessoa* tem associado o seu *nome*, um *email*, uma frase que descreve o seu *estado* e o seu conjunto de *amigos*. Note que as relações de amizade são simétricas, ou seja, se “Ana” é amiga de “Bruno” então “Bruno” é amigo de “Ana”.

Assuma que tipicamente não há mais de 500 pessoas registadas no sistema e que cada um não tem mais de 50 amigos. A aplicação deve permitir:

1. Consultar se uma dada pessoa está registada no sistema (comando `CONSULTAPESSOA`). É fornecido o nome da pessoa. A operação sucede sempre (“Pessoa registada.” ou “Sem registo.”).
2. Registar uma pessoa no sistema (comando `REGISTA`). São fornecidos o nome da pessoa, o email e o estado inicial. Em caso de sucesso, a pessoa é registada (“Pessoa registada com sucesso.”). A operação falha se: (1) já existir uma pessoa registada com esse nome (“Pessoa registada.”)
3. Consultar se uma amizade existe no sistema (comando `CONSULTAAMIZADE`). São fornecidos os nomes das pessoas. A operação sucede sempre retornando a mensagem “Amizade existente.” ou “Amizade inexistente.”
4. Estabelecer uma nova amizade entre duas pessoas (comando `AMIGOS`). São fornecidos os nomes das pessoas. Em caso de sucesso, a amizade é criada (“Amizade criada.”). A operação falha se: (1) um dos nomes não corresponder a uma pessoa registada (“Sem registo.”); (2) a amizade já existir (“Amizade existente.”); ou (3) o registo corresponde a uma relação amizade entre uma única pessoa (“Amizade invalida.”).
5. Listar os amigos de uma pessoa (comando `CONSULTAAMIGOS`). É fornecido o nome da pessoa. Caso não existam amigos é apresentada a mensagem “Nao tem amigos registados.” Caso contrário, a operação apresenta o cabeçalho (“Lista de amigos:”) seguido do nome e email dos amigos. A listagem deve ser ordenada por ordem de inserção. A operação falha se: (1) o nome não corresponder a uma pessoa registada (“Sem registo.”).
6. Modificar o estado de uma pessoa (comando `NOVOESTADO`). São fornecidos o nome da pessoa e o novo estado. Em caso de sucesso o estado é alterado (“Estado alterado.”). A operação falha se: (1) o nome não corresponder a uma pessoa registada (“Sem registo.”).
7. Listar o estado de uma pessoa (comando `CONSULTAESTADO`). É fornecido o nome da pessoa. Em caso de sucesso é apresentado o estado da pessoa. A operação falha se: (1) o nome não corresponder a uma pessoa registada (“Sem registo.”).
8. Listar as pessoas registadas na aplicação (comando `PESSOAS`). A operação sucede sempre. Caso não existam pessoas registadas é apresentada a mensagem “Rede Social vazia.” Caso contrário, a operação apresenta o cabeçalho (“Lista de pessoas registadas:”) seguido do nome e email das pessoas registadas. A listagem deve ser ordenada por ordem de inserção.

9. Terminar a execução do programa (comando **SAIR**). A operação tem sempre sucesso ("Adeus.").

## 2.2 Exemplo de interacção com a aplicação

Desenvolva a sua aplicação para que esta garanta o modelo de interacção ilustrado no exemplo seguinte (o caracter ↵ representa uma mudança de linha):

```
PESSOAS↵
Rede Social vazia.↵
↵
REGISTA↵
Ana↵
ana@gmail.com↵
Bom dia mundo!↵
Pessoa registada com sucesso.↵
↵
REGISTA↵
Ana↵
ana@gmail.com↵
A trabalhar...↵
Pessoa registada.↵
↵
REGISTA↵
Bruno↵
bruno@gmail.com↵
A trabalhar...↵
Pessoa registada com sucesso.↵
↵
CONSULTAAMIZADE↵
Bruno↵
Maria↵
Amizade inexistente.↵
↵
CONSULTAPESSOA↵
Bruno↵
Pessoa registada.↵
↵
CONSULTAPESSOA↵
Daniel↵
Sem registo.↵
↵
AMIGOS↵
Ana↵
Bruno↵
Amizade criada.↵
↵
CONSULTAAMIGOS↵
Bruno↵
Lista de amigos:↵
Ana; ana@gmail.com↵
↵
```

AMIGOS↵  
Ana↵  
Ana↵  
Amizade invalida.↵  
↵  
NOVOESTADO↵  
Ana↵  
Boa tarde!↵  
Estado alterado.↵  
↵  
CONSULTAESTADO↵  
Ana↵  
Boa tarde!↵  
↵  
CONSULTAAMIZADE↵  
Bruno↵  
Ana↵  
Amizade existente.↵  
↵  
CONSULTAESTADO↵  
Daniel↵  
Sem registo.↵  
↵  
NOVOESTADO↵  
Daniel↵  
Surfing...↵  
Sem registo.↵  
↵  
REGISTA↵  
Carlos↵  
carlos@gmail.com↵  
Bem disposto!↵  
Pessoa registada com sucesso.↵  
↵  
AMIGOS↵  
Ana↵  
Carlos↵  
Amizade criada.↵  
↵  
AMIGOS↵  
Ana↵  
Carlos↵  
Amizade existente.↵  
↵  
PESSOAS↵  
Lista de pessoas registadas:↵  
Ana; ana@gmail.com↵  
Bruno; bruno@gmail.com↵  
Carlos; carlos@gmail.com↵  
↵  
CONSULTAAMIGOS↵  
Ana↵  
Lista de amigos:↵  
Bruno; bruno@gmail.com↵  
Carlos; carlos@gmail.com↵  
↵  
SAIR↵  
Adeus.↵  
↵

## 2.3 Desenvolvimento

Desenvolva a sua aplicação de acordo com as seguintes fases:

1. Numa primeira fase, desenvolva o(s) interface(s) de suporte à aplicação *Rede Social*. Faça um esquema para ilustrar os interfaces desenvolvidos.
2. Numa segunda fase implemente e teste a aplicação *Rede Social*.
3. Submeta o código fonte da aplicação ao *Mooshak*.

### Ficheiros de testes

Os testes do Mooshak verificam a implementação dos vários comandos:

- Ficheiro de teste: `1_in_test.txt` (15 pontos)
- Ficheiro de teste: `2_in_test.txt` (15 pontos)
- Ficheiro de teste: `3_in_test.txt` (15 pontos)
- Ficheiro de teste: `4_in_test.txt` (15 pontos)
- Ficheiro de teste: `5_in_test.txt` (15 pontos)
- Ficheiro de teste: `6_in_test.txt` (15 pontos)
- Ficheiro de teste: `7_in_test.txt` (10 pontos)