# Algoritmos e Estruturas de Dados

# SocialNet

## Ano Letivo 2013/2014
## (Versão 30 de Outubro de 2013)

## Alterações (a partir de 11/10/2013):

3.3.3 descrição da operação para a fase 2:

> Fase 2: Nesta fase, o sistema poderá guardar vários contactos. No entanto, cada contacto terá, no máximo, um amigo. A relação de amizade é recíproca. Quando se insere uma amizade com sucesso, nenhum dos contactos envolvidos tem amizades anteriores a esta, ou seja, quando se pode fazer a inserção da amizade no `login1`, também garantidamente se pode fazer a inserção no `login2`.

3.3.5 descrição da operação para as fase 2 e 3:

> Fase 2 e 3: Nestas fases, o sistema poderá guardar vários grupos, e o acesso a cada grupo guardado será conseguido pelo nome do grupo, que identifica o mesmo de forma única.

# 1 Objetivo

O objetivo do trabalho é o desenvolvimento de um Sistema de Gestão de Utilizadores de uma Rede Social com características específicas. Este sistema permite que os seus utilizadores mantenham uma rede de amigos e adiram a um conjunto limitado de grupos de interesse. Os utilizadores podem divulgar as suas opiniões através de Posts, que serão distribuídos pelos amigos e no contexto dos grupos a que pertencem. O sistema irá simular as atividades ligadas à gestão desta rede social, disponibilizando e desenvolvendo todos os Tipos Abstratos de Dados e Estruturas de Dados necessários à criação de um sistema deste tipo. O sistema deverá ser implementado e entregue em 3 fases, o que implicará um desenvolvimento incremental.

# 2 Conceitos e Definições

## 2.1 Contactos, Amigos, *Posts* e Grupos

Existe um conjunto de conceitos associados ao sistema a desenvolver e que se descreve a seguir, de acordo com os requisitos do trabalho.

Um Utilizador é representado na aplicação a desenvolver pelo conceito de *Contacto,* e é identificado por um login (uma cadeia de caracteres sem espaços), tendo também associado, o nome do contacto, a sua idade, a localidade onde habita e a profissão. A cada contacto estão associados os seus *Amigos*, assim como *Posts,* que são mensagens submetidas ou recebidas pelo mesmo. Um post inclui a seguinte informação: Título, Texto e URL para

uma foto relacionada com a mensagem. Um determinado contacto pode ainda pertencer a um ou mais grupos (no máximo 10) existentes na rede.

Um *Grupo* é identificado pelo seu Nome e tem associado um texto com a sua descrição, que reflete o tema subjacente ao grupo. Tal como um contacto, também o grupo tem contactos associados, que constituem os seus *Aderentes*. Cada grupo deve ainda aceder a todos os posts dos seus aderentes.

## 2.2 Gestão Amizades/Aderências

Nesta rede social os pedidos de amizade ou de aderência a um grupo são sempre aceites ☺. Além disso, esta relação é recíproca, ou seja se o João é amigo do Luís, o Luís vai retribuir essa amizade. A partir do momento em que a amizade se inicia, o Luís vai receber, na sua zona de comunicação, todos os posts do João e vice-versa. Se a amizade terminar, estes posts não são removidos da zona de comunicação dos perfis, apenas não serão adicionados novos.

O mesmo acontece com a gestão das aderências a grupos. Quando um contacto adere a um grupo, os seus posts passam a ser adicionados à zona de comunicação do grupo, mas os mesmos posts não são removidos dessa zona, caso o contacto decida abandonar o grupo.

Quando um contacto faz um post, este é então enviado a todos os seus amigos e é também inserido na sua própria zona de comunicação. Os grupos a que o contacto aderiu também recebem o mesmo post. Os posts são guardados por ordem de receção, do mais recente para o mais antigo.

Os aderentes de um grupo não recebem os posts do grupo na sua zona de comunicação, mas podem consultar os mesmos, quando acedem ao grupo. Assim, a consulta da zona de comunicação de um grupo só é permitida aos seus aderentes.

# 3 Especificação do Sistema

O sistema deverá ler comandos da entrada padrão (`System.in`), processando-os um a um e enviando os resultados para a saída padrão (`System.out`). A terminação ocorrerá quando for atingido o fim de ficheiro na entrada padrão. A execução do programa pode ser assegurada com a introdução de dados e respetiva apresentação de resultados em ficheiro, através do redireccionamento do input e do output. Este processo é explicado na página "Recomendações para os testes do Trabalho" que será disponibilizada atempadamente na página da disciplina no Moodle.

A persistência dos dados deve ser assegurada entre execuções consecutivas. Isto significa que, antes de terminar a execução, o sistema deve guardar o estado da base de dados, utilizando as funcionalidades de serialização do Java. Na próxima execução do programa, os dados armazenados deverão ser carregados e o estado do sistema reconstituído.

## 3.1 Sintaxe

Pretende-se que a interface da aplicação seja muito simples, de modo a poder ser utilizada em ambientes diversos e, no caso da saída, para permitir automatizar o processo de teste.

Por estes motivos, a entrada e a saída deverão respeitar o formato rígido que se indica na Secção 3.3. Convém referir que o símbolo ↵ representa uma mudança de linha e que cada comando termina com duas mudanças de linha.

Poderá admitir que a entrada está sempre sintaticamente correta e que os dados satisfazem as restrições enunciadas na secção 3.2.

## 3.2 Tipos dos Dados e dos resultados

Os campos identificativos de contacto (login) e grupo (nome) são cadeias de 9 caracteres, sem espaços. Os nomes de contactos têm 50 caracteres e todos os textos, assim como os URLs, têm 200 caracteres. As restantes cadeias de caracteres terão um comprimento de 20. A idade dos utilizadores é um inteiro. As cadeias de caracteres de dados a inserir no sistema não deverão conter mudanças de linha (símbolo ↵).

A introdução de dados poderá ser feita em minúsculas ou em maiúsculas, desde que as comparações entre sequências de caracteres sejam independentes da forma como a introdução for feita. Por exemplo, para a aplicação, o utilizador "Joao A" deverá ser igual a "JOAO A" e a "joao a". O output dos dados contidos no sistema deverá ser sempre feito em maiúsculas.

## 3.3 Operações a implementar

Estas operações serão desenvolvidas incrementalmente. Assim, a descrição de cada uma das operações poderá ser diferente para cada uma das fases do trabalho a implementar, ou não. Em cada uma das subsecções abaixo, cada operação é especificada, de acordo com as diferenças por fase.

### 3.3.1 Inserção de um contacto

- SINTAXE DE ENTRADA

  **IC** *login nomeContacto*↵

  *idade localidade*↵

  *profissao*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Inserção de um contacto identificado pelo login ***login*** e relativo ao contacto com o nome ***nomeContacto***, com a idade ***idade,*** residente em ***localidade*** e com a profissão ***profissao***.

  Fase 1: Nesta fase o sistema só guarda os dados de um único contacto. Se o utilizador tentar inserir um novo contacto quando já existe um no sistema, o mais antigo será apagado.

Fases 2 e 3: Nestas fases o sistema poderá guardar vários contactos, e o acesso a cada contacto guardado será conseguido por login. O login identifica o contacto de forma única.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  `Insercao de contacto com sucesso.`↵

  ↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  `Existencia de contacto referido.`↵

  ↵

### 3.3.2 Consultar os dados de um contacto

- SINTAXE DE ENTRADA

  **CC** *login*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Consulta dos dados de um contacto identificado por ***login***.

  Fase 1: Nesta fase, o sistema só guarda os dados de um único contacto.

  Fases 2 e 3: Nestas fases, o sistema poderá guardar vários contactos, e a pesquisa do contacto a consultar será efetuada pelo seu login, que deve ser único.

SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

*login nomeContacto idade*↵

*localidade profissao*↵

↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  `Inexistencia de contacto referido.`↵

  ↵

### 3.3.3 Inserção de amizade

- SINTAXE DE ENTRADA

  **IA** *login1 login2*↵

↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Inserção de uma amizade entre o contacto com o login ***login1*** e o contacto com o login ***login2***.

  Fase 1: Nesta fase o sistema só guarda os dados de um contacto, pelo que não deverão ser inseridas amizades. Implicitamente, um contacto é amigo de si próprio.

  Fase 2: Nesta fase, o sistema poderá guardar vários contactos. No entanto, cada contacto terá, no máximo, um amigo. A relação de amizade é recíproca. Quando se insere uma amizade com sucesso, nenhum dos contactos envolvidos tem amizades anteriores a esta, ou seja, quando se pode fazer a inserção da amizade no ***login1***, também garantidamente se pode fazer a inserção no ***login2***.

  Fase 3: Nesta fase, o sistema poderá guardar vários contactos e cada contacto poderá ter vários amigos. Não esquecer que a relação de amizade é recíproca.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  `Insercao de amizade com sucesso.`↵

  ↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  *mensagem-de-inserção-de-amizade*↵

  ↵

- A *mensagem-de-inserção-de-amizade* é uma das seguintes:
  - `Inexistencia de contacto referido.`
    Quando um dos logins referidos não existe no sistema.
  - `Existencia de amizade referida.`
    Quando já existe amizade entre os contactos com os logins inseridos.

### 3.3.4 Remoção de amizade

- SINTAXE DE ENTRADA

  **RA** *login1 login2*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Fim da amizade entre os contactos com os logins ***login1*** e ***login2***.

  Fases 1 e 2: Nestas fases, cada contacto possui, no máximo, um amigo, sendo que na fase 1 essa amizade é, implicitamente consigo próprio. A amizade consigo

próprio não pode ser removida. Na fase 2, se a amizade for removida, o contacto fica sem amigos.

Fase 3: Nesta fase, cada contacto poderá ter vários amigos, e a remoção de uma amizade é efetuada através dos logins dos contactos, que são únicos.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  `Remocao de amizade com sucesso.`↵

  ↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  *`mensagem-de-remoção-de-amizade.`*↵

  ↵

- A *`mensagem-de-remoção-de-amizade`* é uma das seguintes:
  - `Inexistencia de contacto referido.`
    Quando um dos logins referidos não existe no sistema.
  - `Inexistencia de amizade referida.`
    Quando não existe amizade entre os contactos com os logins inseridos.
  - `Amizade nao pode ser removida.`
    Quando o contacto tenta remover a amizade consigo próprio.

### 3.3.5 Inserção de grupo

- SINTAXE DE ENTRADA

  **`IG`** *`nomeGrupo`*↵

  *`textoDescritivo`*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Inserção de um grupo identificado por *`nomeGrupo`*, descrito por *`textoDescritivo`*.

  Fase 1: Nesta fase o sistema só guarda os dados de um grupo. Se o utilizador tentar inserir um novo grupo quando já existe um no sistema, o mais antigo será sempre apagado.

  Fase 2 e 3: Nestas fases, o sistema poderá guardar vários grupos, e o acesso a cada grupo guardado será conseguido pelo nome do grupo, que identifica o mesmo de forma única.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  ```
  Insercao de grupo com sucesso.↵
  ↵
  ```

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  ```
  Existencia de grupo referido.↵
  ↵
  ```

### 3.3.6 Consultar os dados de um grupo

- SINTAXE DE ENTRADA

  **CG** *nomeGrupo*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Consulta dos dados de um grupo identificado por *nomeGrupo*. Os dados a consultar são o nome do grupo e a sua descrição.

  Fase 1: Nesta fase, o sistema só guarda os dados de um único grupo.

  Fases 2 e 3: Nestas fases, o sistema poderá guardar vários grupos, e a pesquisa do grupo a consultar será efetuada pelo seu nome, que deve ser único.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  *nomeGrupo*↵

  *textoDescritivo*↵

  ↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  ```
  Inexistencia de grupo referido.↵
  ↵
  ```

### 3.3.7 Remoção de grupo

- SINTAXE DE ENTRADA

  **RG** *nomeGrupo*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Remoção de grupo identificado pelo nome ***nomeGrupo***. O grupo a remover deve ser ainda removido dos grupos de todos os aderentes ao grupo.

  Fase 1: Nesta fase, existe apenas um grupo.

  Fase 2: Nesta fase, existem vários grupos no sistema, cada um identificado pelo seu nome, que é único. Cada grupo pode ter apenas um contacto aderente.

  Fase 3: Nesta fase, além de existirem vários grupos no sistema, cada grupo pode ter vários contactos aderentes.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  ```
  Remocao de grupo com sucesso.↵

  ↵
  ```

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  ```
  Inexistencia de grupo referido.↵

  ↵
  ```

### 3.3.8 Inserção de aderente de grupo

- SINTAXE DE ENTRADA

  **ID** *login nomeGrupo*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Aderência de contacto com o login ***login*** ao grupo ***nomeGrupo***. Em caso de sucesso, o grupo deverá ser adicionado aos grupos do contacto e o contacto deverá ser adicionado aos aderentes do grupo.

  Fase 1: Nesta fase o sistema só guarda os dados de um contacto e de um grupo. Se o utilizador tentar inserir uma aderência a um grupo quando o grupo já tem um aderente, o mais antigo será apagado.

  Fase 2: Nesta fase, o sistema poderá guardar vários contactos e vários grupos. Um contacto poderá aderir a vários grupos mas cada grupo terá, no máximo, um aderente.

  Fase 3: Nesta fase, o sistema poderá guardar vários contactos e vários grupos. Um contacto poderá aderir a vários grupos e cada grupo poderá ter vários aderentes.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

`Insercao de aderencia a grupo com sucesso.`↵

↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  *mensagem-de-inserção-de-aderência*↵

  ↵

- A *mensagem-de-inserção-de-aderência* é uma das seguintes:
  - `Inexistencia de contacto referido.`
    Quando o login referido não existe no sistema.
  - `Inexistencia de grupo referido.`
    Quando o nome de grupo referido não existe no sistema.
  - `Existencia de aderencia referida.`
    Quando o contacto referido já aderiu ao grupo referido.

### 3.3.9 Remoção de aderente de grupo

- SINTAXE DE ENTRADA

  **RD** *login nomeGrupo*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Remoção de aderente com o login *login* do grupo com o nome *nomeGrupo*. Esta operação implica a remoção do contacto dos aderentes do grupo. O grupo deverá ainda ser removido dos grupos do contacto.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  `Remocao de aderencia com sucesso.`↵

  ↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  *mensagem-de-remoção-de-aderência.*↵

  ↵

- A *mensagem-de-remoção-de-aderência* é uma das seguintes:
  - `Inexistencia de contacto referido.`
    Quando o login referido não existe no sistema.

- `Inexistencia de grupo referido.`
  Quando o nome de grupo referido não existe no sistema.
- `Inexistencia de aderencia referida.`
  Quando o contacto referido não aderiu ao grupo referido.

### 3.3.10 Inserção de post de contacto

- SINTAXE DE ENTRADA

  **IP** *login*↵

  titulo↵

  textoDescritivo↵

  *URL*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Inserção de post na zona de comunicação do contacto com o login ***login***. Esta inserção implica também a inserção do post na zona de comunicação de todos os amigos do contacto, assim como em todos os grupos a que o contacto aderiu. Um utilizador pode submeter vários posts.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  Insercao de post com sucesso.↵

  ↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  Inexistencia de contacto referido.↵

  ↵

### 3.3.11 Listar amigos de um contacto

- SINTAXE DE ENTRADA

  **LA** *login*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Listagem, por ordem lexicográfica do login, dos amigos do contacto com login ***login***. A listagem apresenta todos os amigos do contacto, diferentes dele próprio.

Fases 1 e 2: Nestas fases, cada contacto possui, no máximo, um amigo. Um contacto é, implicitamente, amigo de si mesmo.

Fase 3: Nesta fase, cada contacto poderá ter vários amigos.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  *Contacto-Amigo*↵

  *Contacto-Amigo*↵

  *Contacto-Amigo*↵

  *Contacto-Amigo*↵

  . . . . . .

  ↵

- *Contacto-Amigo* tem o seguinte formato:

  *login nomeContacto.*

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  *mensagem-de-listagem-de-amigos*↵

  ↵

- A *mensagem-de-listagem-de-amigos* é uma das seguintes:
  - `Inexistencia de contacto referido.`
    Quando não existe um contacto com o login referido.
  - `Contacto nao tem amigos.`
    Quando o contacto não tem amigos (excetuando o próprio).

### 3.3.12 Listar aderentes de um grupo

- SINTAXE DE ENTRADA

  **LD** *nomeGrupo*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Listagem, por ordem lexicográfica do login, dos aderentes de um grupo.

  Fases 1 e 2: Nestas fases, cada grupo possui, no máximo, um aderente.

Fase 3: Nesta fase, cada grupo poderá ter vários aderentes.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  *Contacto-Aderente*↵

  *Contacto-Aderente*↵

  *Contacto-Aderente*↵

  *Contacto-Aderente*↵

  . . . . . .

  ↵

- *Contacto-Aderente* tem o seguinte formato:

  *login nomeContacto.*

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  *mensagem-de-listagem-de-aderentes*↵

  ↵

- A *mensagem-de-listagem-de-aderentes* é uma das seguintes:
  - `Inexistencia de grupo referido.`
    Quando não existe um grupo com o nome referido.
  - `Grupo nao tem aderentes.`
    Quando o grupo não tem aderentes.

### 3.3.13 Listar posts de um contacto

- SINTAXE DE ENTRADA

  **LC** *login1 login2*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Pedido de listagem, por ordem de inserção, do mais recente para o mais antigo, dos posts publicados na zona de comunicação do contacto com login *login1*, submetido pelo contacto com login *login2*. Este pedido só terá sucesso se os contactos referidos forem amigos.

Fase 1: Nesta fase, o sistema contém apenas um contacto, pelo que este pedido só terá sucesso se for submetido pelo próprio.

Fase 2: Nesta fase, um contacto tem, no máximo um amigo.

Fase 3: Nesta fase, um contacto pode ter mais de um amigo.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  *Post-contacto*↵

  *Post-contacto*↵

  *Post-contacto*↵

  *Post-contacto*↵

  . . . . . .

  ↵

- *Post-contacto* tem o seguinte formato:

  *titulo*↵

  *textoDescritivo*↵

  *URL*↵

  ↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  *mensagem-de-listagem-de-posts-de-amigo*↵

  ↵

- A *mensagem-de-listagem-de-posts-de-amigo* é uma das seguintes:
  - `Inexistencia de contacto referido.`
    Quando não existe um contacto com o login referido.
  - `Contacto nao tem permissao de leitura de posts.`
    Quando os contactos referidos não são amigos.
  - `Contacto nao tem posts.`
    Quando o contacto referido não tem posts.

### 3.3.14 Listar posts de um grupo

- SINTAXE DE ENTRADA

  **LG** *nomeGrupo login*↵

  ↵

- DESCRIÇÃO DA OPERAÇÃO (SE EFECTUADA COM SUCESSO)

  Pedido de listagem, por ordem de inserção, do mais recente para o mais antigo, dos posts publicados na zona de comunicação do grupo com nome *nomeGrupo*, submetido pelo contacto com login *login*. Este pedido só terá sucesso se o contacto referido for aderente do grupo referido.

  Fase 1: Nesta fase, o sistema contém apenas um contacto e um grupo.

  Fase 2: Nesta fase, um grupo tem, no máximo, um aderente.

  Fase 3: Nesta fase, um grupo pode ter mais de um aderente.

- SINTAXE DE SAÍDA – OPERAÇÃO EFECTUADA COM SUCESSO

  *Post-grupo*↵

  *Post-grupo*↵

  *Post-grupo*↵

  *Post-grupo*↵

  . . . . . .

  ↵

- *Post-grupo* tem o seguinte formato:

  *titulo*↵

  *textoDescritivo*↵

  *URL*↵

  ↵

- SINTAXE DE SAÍDA – OPERAÇÃO NÃO EFECTUADA

  *mensagem-de-listagem-de-posts-de-grupo*↵

  ↵

- A *mensagem-de-listagem-de-posts-de-grupo* é uma das seguintes:
  - `Inexistencia de grupo referido.`
    Quando não existe um grupo com o nome referido.
  - `Inexistencia de contacto referido.`
    Quando não existe um contacto com o login referido.
  - `Contacto nao tem permissao de leitura de posts.`
    Quando o contacto referido não aderiu ao grupo.

- `Grupo nao tem posts.`
  Quando o grupo não tem posts.

## 3.4 Números esperados

Relativamente à fase final do trabalho, não existem restrições quanto ao número de contactos ou de grupos que podem existir numa determinada altura. O número de amigos ou de posts associados a um contacto também não é limitado superiormente mas, cada contacto pode aderir, no máximo, a 10 grupos. Também o número de aderentes e de posts de um grupo não têm limites. Poderemos, no entanto, prever um número de 12 000 contactos e 3 000 grupos. Cada contacto poderá gerar 200 posts por ano.

## 3.5 Requisitos do Mooshak

### 3.5.1 Regras a Satisfazer pelo Programa

Para submeter o trabalho ao Mooshak, é necessário que o programa fonte respeite as quatro regras seguintes:

- O código fonte completo (ficheiros *.java) tem de ser guardado num único arquivo ZIP;
- A classe principal tem de se chamar Main e tem de estar na raiz do arquivo (i.e., tem de pertencer ao pacote principal (*default*);
- As pastas correspondentes aos pacotes do trabalho têm de estar na raiz do arquivo;
- O programa só pode criar ficheiros na directoria corrente.

### 3.5.2 Como preparar o arquivo ZIP

Para evitar problemas causados pela dimensão do ficheiro submetido, somente o código fonte (ficheiros *.java) deve ser enviado para o servidor.

Assuma, para exemplificar, que o código fonte do trabalho está dentro de uma pasta chamada `/home/me/aulas/AED/trabalhoFinal/src` e que, dentro dessa pasta, há um ficheiro e duas sub-pastas.

```
Main.java
dataStructures
socialNet
```

Nesse caso, o arquivo poderia ser construído (em Linux) com o seguinte comando:

```
zip aed.zip Main.java dataStructures/*.java socialNet/*.java
```

executado na pasta `/home/me/aulas/AED/trabalhoFinal/src`.

# 4 Desenvolvimento

O trabalho é realizado em grupos de dois alunos, do mesmo turno prático (ou em dois turnos práticos lecionados pelo mesmo docente, se este autorizar – o docente do seu turno prático deve ser o contacto em tudo o que envolva a constituição de grupos de trabalho). O

trabalho é implementado em Java, nomeadamente em JDK 1.6, instalado nos laboratórios das aulas práticas, em Windows e Linux.

Há duas versões do trabalho, tal como descrito abaixo:

- Na **Versão I**, avaliada entre 10 e 20 valores, não é permitido fazer uso do *package java.util*, à exceção da classe *Scanner*;
- Na **Versão J**, avaliada entre 10 e 15 valores, podem ser usadas todas as interfaces e classes do Java.

## 4.1 Entregas e Faseamento

Como já descrito acima, o trabalho será desenvolvido incrementalmente, em três fases diferentes, com entregas marcadas. Em cada fase, todas as operações deverão ser implementadas, mas poderão crescer de complexidade de fase para fase, tal como descrito na secção 3.3.

A entrega no Mooshak, é constituída por um zip contendo o código fonte **devidamente comentado**, tal como descrito na secção 3.5.2. O nome e número dos alunos que compõem o grupo deverá ser inserido em todos os ficheiros entregues.

## 4.2 Entrega final

Adicionalmente, na última fase deverá ainda ser entregue, no Moodle e em papel (na secretaria do DI) um relatório final do trabalho, contendo o seguinte:

- Para cada um dos Tipos Abstratos de Dados (Interfaces) definidos no trabalho, uma justificação para as implementações realizadas para os mesmos, especificamente para as estruturas de dados escolhidas;
- Para cada uma das operações descritas em 3.3, uma descrição do comportamento do trabalho, em termos das operações efetuadas sobre as estruturas de dados;
- Uma versão impressa do código fonte e do diagrama de classes e interfaces;
- O estudo das complexidades temporais das operações descritas em 3.3, no melhor caso, no pior caso e no caso esperado;
- O estudo da complexidade espacial da solução proposta.

# 5 Avaliação

O trabalho é obrigatório para todos os alunos que não têm frequência e dá frequência. A avaliação incidirá sobre todos os aspetos: conceção, qualidade e eficiência da solução, modularidade, estrutura e documentação do código, qualidade do relatório final, etc. As fases 1 e 2 valem ambas 5% da nota final da disciplina e a fase 3 vale 15% da mesma nota. Se uma das fases não for entregue dentro do prazo definido, a nota associada à mesma fase será 0 (zero).

## 5.1 Testes e discussão

O trabalho terá de satisfazer todos os requisitos especificados na secção 3. Essa verificação será efetuada automaticamente pelo sistema Mooshak, com os Testes de Avaliação.

Em cada fase:

- Se o programa não passar todos os Testes de Avaliação, os seus autores obterão a nota de 0 (zero) na fase em questão;
- Se o programa passar todos os Testes de Avaliação da fase, o docente do turno prático dos alunos que constituem o grupo fará uma avaliação preliminar da fase, que deverá ser confirmada no final do semestre, numa discussão final.

No final do semestre, todos os grupos em posição de obter frequência serão submetidos a uma discussão do trabalho, para verificação de autoria. Nesta discussão, os membros do grupo poderão ser questionados sobre **qualquer das classes e interfaces submetidas ao Mooshak em qualquer das fases de submissão**.

Se se detetar:

- Que um trabalho não foi realizado apenas pelos alunos que o assinaram;
- Que um aluno assinou um trabalho que não realizou; ou
- Que a distribuição das tarefas pelos membros do grupo não foi equilibrada,

esse trabalho será anulado e **nenhum** dos elementos do(s) grupo(s) envolvido(s) obterá frequência.

Se um aluno faltar à discussão do trabalho que assinou, não obterá frequência, reprovando à disciplina.

## 5.2 Resumo das datas importantes

- Submissão da fase 1 do trabalho, no Mooshak: entre 7 e 11 de Outubro de 2013. No dia 11 de Outubro a entrega eletrónica tem de ser realizada até às **17h**.
- Submissão da fase 2 do trabalho, no Mooshak: entre 4 e 8 de Novembro de 2013. No dia 8 de Novembro a entrega eletrónica tem de ser realizada até às **17h**.
- Submissão da fase 3 do trabalho e do relatório final, no Mooshak e Moodle: entre 25 e 29 de Novembro de 2013. No dia 29 de Novembro a entrega eletrónica tem de ser realizada até às **17h**.
- Marcação das discussões finais: 2 de Dezembro de 2013.
- Entrega em papel na secretaria do DI: até às **16h** de 29 de Novembro de 2013.
- Realização das discussões: entre 3 e 10 de Dezembro, **em princípio,** dentro do horário da disciplina (da aulas teóricas ou práticas).