

TAD Graph – Rede Social (avaliado)

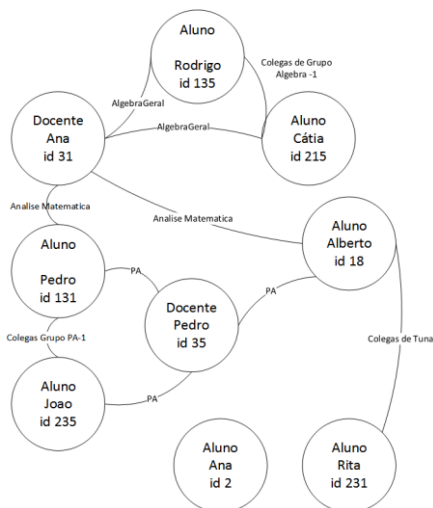
Objetivos:

- Utilização do TAD Graph e manipulação da estrutura de dados da implementação fornecida.

Introdução:

Comece por clonar o projeto existente no GitHub no seguinte endereço:

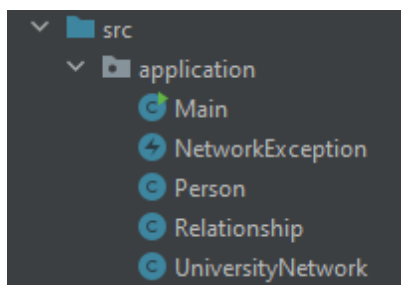
<https://github.com/estsetubal-pa-geral/Lab6Template.git>



A classe **Person** serve para representar as pessoas da rede social. A classe **Relationship** serve para representar o relacionamento estabelecido entre as pessoas da ESTSetubal.

Figura 1 – Exemplo de uma rede social universitária.

Verifique que já existem criadas as classes presentes na figura:



O objetivo do laboratório é fazer a implementação da classe **UniversityNetwork** que utiliza o ADT Graph<V, E> para implementar uma rede social da EST Setubal.

A classe *UniversityNetwork* é responsável por gerir uma rede social de uma escola e contempla uma instância de *Graph* com nome *network*. Esta rede é composta pelas pessoas e pelas relações estabelecidas entre estas.

Existe, ainda, um método privado **findPerson** que devolve o vértice com a pessoa com um determinado **id**.

Nível I

1. Implemente na classe *UniversityNetwork* os seguintes métodos públicos:

- **void addPerson(Person p)**
 throws NetworkException

Adiciona uma pessoa à rede. Se a pessoa já existir, lança uma exceção.

- **void addGroupRelationship (String description, int idStudent1, int idStudent2)**
 throws NetworkException

Adiciona uma relação do tipo **RelRole.GROUP**. Recebe o id dos dois estudantes envolvidos, assim como a descrição (description) da relação a criar. Deverá lançar uma exceção se os ids não existirem ou não se referirem a pessoas do tipo correto.

- **void addClassRelationship (String description, int idTeacher, int idStudent)**
 throws NetworkException

Adiciona uma relação do tipo **RelRole.CLASS**, recebe o id do estudante e o id do professor sobre o qual se vai estabelecer a relação, assim como a descrição (description) da mesma. Deverá lançar uma exceção se os ids não existirem ou se não forem pessoas do tipo correto.

2. No método *main()*, utilizando os métodos anteriores, crie a rede ilustrada na Figura 1; A descrição da relação corresponde ao texto ilustrado na aresta.

Nível II

1. Crie o seguinte método na classe *UniversityNetwork*:

- **boolean personExists(String name)**
 Informa se existe uma determinada pessoa como vértice.

Testes unitários:

2. Comece por criar o método **setup** de modo a disponibilizar uma configuração inicial a ser utilizada por cada um dos testes unitários a implementar.

3. De seguida, implemente os seguintes testes unitários:

- **test_personDoesNotExists_whenEmptyGraph()**
 Verifica que não existe a pessoa com o nome "Rodrigo" com o grafo vazio.
- **isThrown_Exception_AfterAddExistentPerson()**
 Verifica se é lançada exceção após adicionar uma pessoa já existente.
- **test_personExists_afterInsert()**
 Verifica que a pessoa com o nome "Rodrigo" existe após ser adicionada.

Nível III

1. Implemente na classe **UniversityNetwork** os seguintes métodos públicos:

- **List<Person> getIsolated()**
Devolve uma lista com as pessoas que se encontram isoladas, ou seja que não têm relações com ninguém.
- **List<Relationship> getRelationship(int id1, int id2)**
throws NetworkException
Dados os ids de duas pessoas, devolve uma lista contendo as relações existente entre estas. Deverá ser lançada uma exceção no caso de qualquer dos ids não for válido.

2. Adicione o seguinte teste unitário:

- **isThrown_Exception_AfterGetInvalidRelationship()** – Verifica se é lançada exceção após procurar a relação entre duas pessoas em que um dos ids não é válido.

Nível IV

Implemente na classe **UniversityNetwork** o método **String toStringTeachersStudents()** que devolve uma representação textual de cada um dos docentes e respetivos alunos, tal como ilustrado na Figura 2.

```
Docentes
Professor Pedro(35)
  de PA de Aluno Alberto(18)
  de PA de Aluno Pedro(131)
  de PA de Aluno Joao(235)
Professor Ana(31)
  de Algebra Geral de Aluno Rodrigo(135)
  de Algebra Geral de Aluno Catia(215)
  de Análise Matematica de Aluno Alberto(18)
  de Análise Matematica de Aluno Pedro(131)
```

Figura 2

Teste o método anterior.

Nível V

Implemente na classe **UniversityNetwork** os seguintes métodos e teste-os no main():

- **int getNumberOfStudents(int id) throws NetworkException**
retorna o número de estudantes que um professor com um determinado id leciona. Lança uma exceção se o id não existir ou se não for de um docente.
- **Person getPopular()**
retorna a pessoa mais popular da rede (a que tem mais relacionamentos).
- **void removeRelation(int id1, int id2) throws NetworkException**
remove todos os relacionamentos existentes entre as pessoas com id1 e id2. Lança uma exceção se não existir nenhum relacionamento entre as pessoas especificadas.