

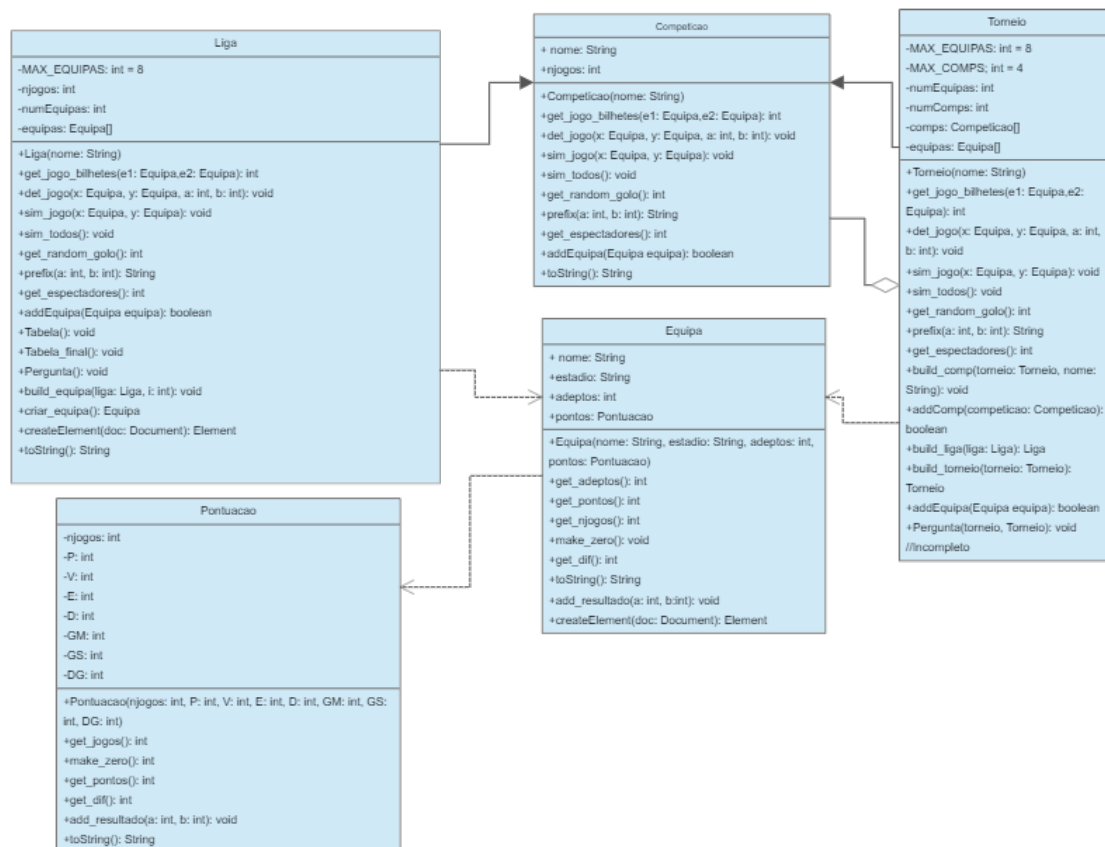


Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia

Modelação e Programação - MP - 2122SV

Trabalho Prático 4



Docente Carlos Júnior
Realizado por :
Pedro Silva 48965

6 de junho de 2022

Conteúdo

1	Introdução	I
2	Requisitos	I
3	UML	I
4	Wireframe da interface gráfica do simulador da Liga	II
5	DTD	II
6	Classes	III
6.1	Pontuacao	III
6.2	Equipa	III
6.3	Liga	III
6.4	Competicao	IV
6.5	Torneio	V
7	Conclusão	V

Lista de Figuras

1	UML da aplicação Java	I
2	Wireframe simulador Liga	II
3	DTD da aplicação Java	II

1 Introdução

O TP4 pretende consolidar e avaliar os conhecimentos adquiridos em Modelação e Programação tendo como objetivo o desenvolvimento de uma aplicação usando a linguagem Java com interface gráfica. A aplicação desenvolvida neste trabalho tem o objetivo de simular uma competição de futebol podendo esta ser um torneio ou uma liga e armazenar a sua informação num ficheiro ".xml".

2 Requisitos

Os objetivos deste trabalho são:

- Desenhar o UML da aplicação;
- Conceber a aplicação Java;
- Integrar o armazenamento de dados na aplicação;
- Planear a gramática (DTD) e um documento XML;

3 UML

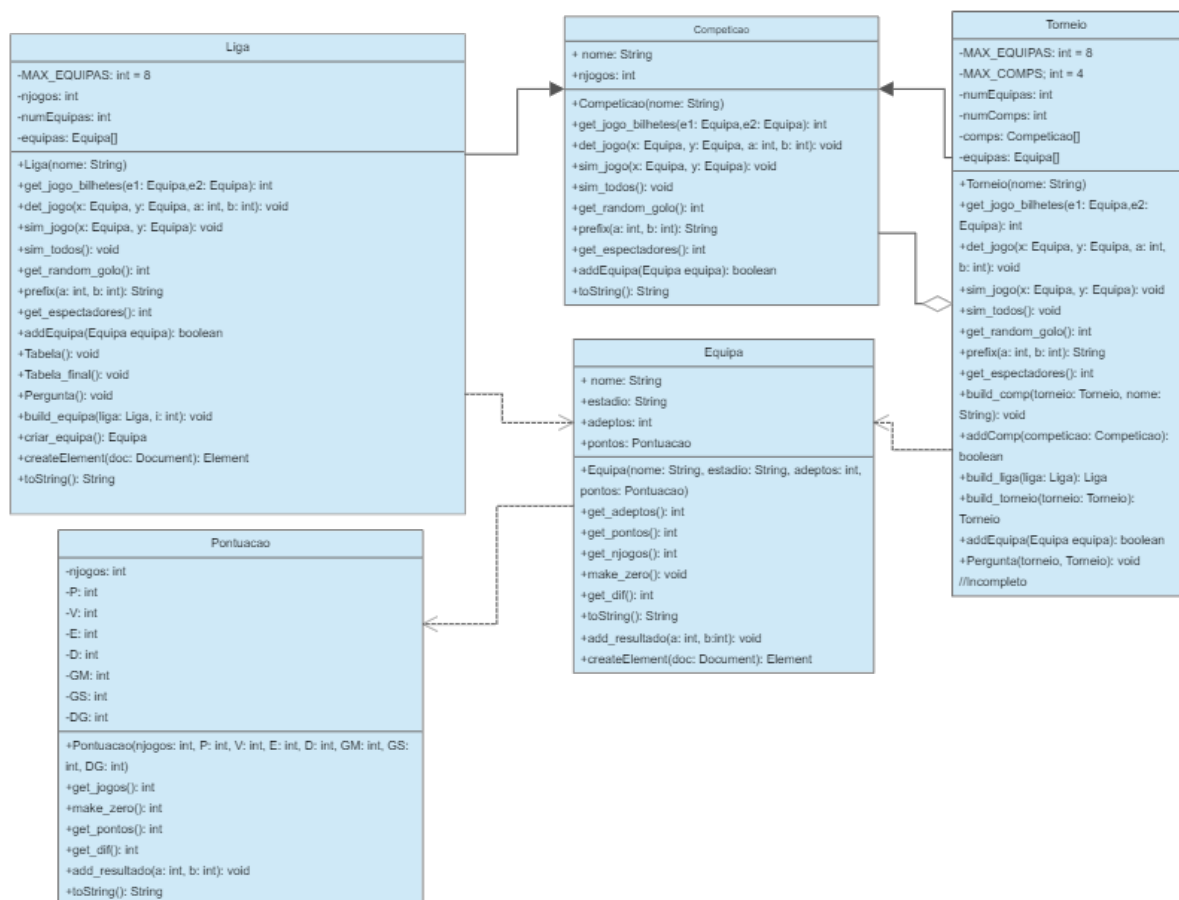


Figura 1: UML da aplicação Java

4 Wireframe da interface gráfica do simulador da Liga

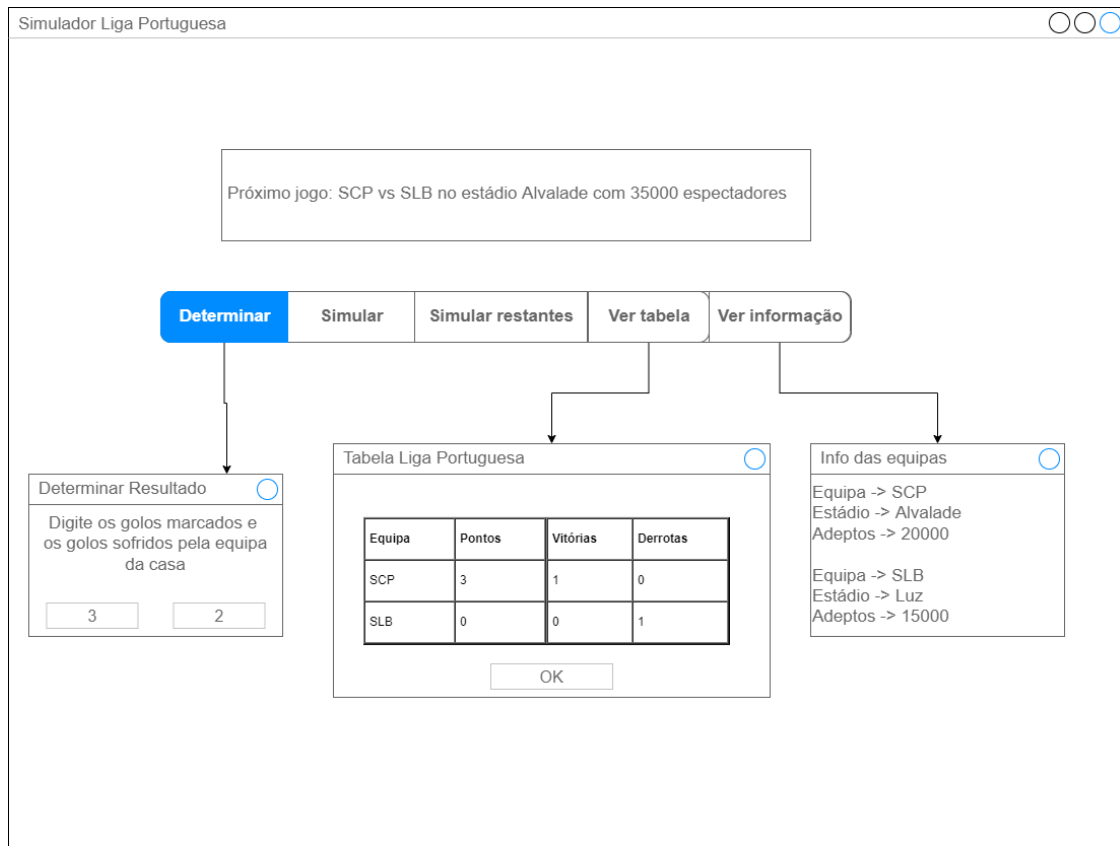


Figura 2: Wireframe simulador Liga

5 DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE BaseDados [
  <!ELEMENT BaseDados (Torneios|Equipas|Ligas)*>
  <!ELEMENT Torneios (#PCDATA)>
  <!ELEMENT Equipas (Equipa)*>
  <!ELEMENT Equipa (Nome|Estadio)*>
  <!ATTLIST Equipa
    Adeptos CDATA #REQUIRED>
  <!ELEMENT Nome (#PCDATA)>
  <!ELEMENT Estadio (#PCDATA)>
  <!ELEMENT Ligas (Liga)*>
  <!ELEMENT Liga (Nome|Equipa)*>
  <!ATTLIST Liga
    Comp CDATA #REQUIRED>
]>
```

Figura 3: DTD da aplicação Java

6 Classes

6.1 Pontuacao

Tem como objetivo guardar as informações sobre a classificação da equipa. O construtor recebe o número de jogos, vitórias, empates e derrotas e estes têm que ser maior que 0 e menor que 14 (o número total de jogos realizados por cada equipa). A soma das vitórias, empates e derrotas tem que ser igual ao número de jogos se não lança exceção. Também recebe o número de golos marcados e sofridos e a sua diferença. Os métodos desta classe são:

- "get_jogos" devolve o número de jogos;
- "make_zero" faz com que a informação sobre os golos da Pontuacao fique igual a 0;
- "get_pontos" devolve os pontos;
- "get_dif" devolve a diferença de golos;
- "add_resultado" adiciona o resultado do jogo à pontuação;
- "toString" retorna uma String da pontuação formatada de acordo com a tabela;

6.2 Equipa

Tem como objetivo criar uma equipa e atribuir-lhe uma pontuação. O construtor recebe a sigla de 3 caracteres da equipa, o nome do estádio, o número de adeptos e a Pontuacao. Os métodos desta classe são:

- "get_adeptos" devolve o número de adeptos;
- "get_jogos" devolve o número de jogos;
- "make_zero" faz com que a informação sobre os golos da Pontuacao fique igual a 0;
- "get_pontos" devolve os pontos;
- "get_dif" devolve a diferença de golos;
- "add_resultado" adiciona o resultado do jogo à pontuação;
- "toString" retorna uma String com o nome da equipa e o nome do estádio;
- "createElement" devolve um novo elemento criado a partir da equipa recebida;

6.3 Liga

Tem como objetivo criar uma liga que vai conter 8 equipas seleccionadas ou não pelo utilizador e depois de criada armazenar no ficheiro xml. É classe filha da classe Competicao o que torna possível o uso de "super()" que chama o método da classe pai. O construtor recebe o nome da liga. Os métodos desta classe são:

- "get_random_golo" chama a superclasse com o mesmo nome;
- "det_jogo" chama a superclasse com o mesmo nome;;
- "sim_jogo" chama a superclasse com o mesmo nome;

- "sim_todos" simula os restantes jogos utilizando o método "sim_jogo";
- "prefix" chama a superclasse com o mesmo nome;
- "get_espectadores" devolve a soma do número de espetadores de todas as equipas da liga;
- "addEquipa" adiciona a equipa recebida à liga;
- "get_jogo_bilhetes" chama a superclasse com o mesmo nome;
- "Tabela" imprime na consola a tabela classificativa da liga;
- "Tabela_final" imprime na consola a tabela classificativa da liga e organiza o array da liga de acordo com a classificação;
- "Pergunta" pergunta ao utilizador se quer criar novas equipas para adicionar à liga;
- "build_equipa" adiciona à liga equipas previamente criadas a partir de um ficheiro xml utilizando o XPath;
- "criar_equipa" Retorna uma equipa criada pelo utilizador;
- "toString" chama a superclasse com o mesmo nome;
- "createElement" devolve um novo elemento criado a partir da liga recebida atribuindo-lhe a identificação "L" de liga;

6.4 Competicao

É a classe pai da classe "Liga" e "Torneio" tendo como objetivo facilitar o trabalho do programador ao fazer com que este possa escrever código apenas uma vez e chamar sempre que precisa o método nas classes filho. Também possui classes abstratas que servem como guia para as classes filho, ou seja, estas têm que implementar um método para cada classe abstrata. O construtor recebe o nome da competição.

- "get_random_golo" escolhe aleatoriamente o número de golos marcado pela equipa tendo maior probabilidade o número 0 e 1;
- "det_jogo" adiciona o resultado do jogo fornecido pelo utilizador;
- "sim_jogo" adiciona um resultado aleatório fornecido pelo método "get_random_golo";
- "sim_todos" classe abstrata;
- "prefix" devolve em String o resultado do jogo fornecido;
- "get_espectadores" classe abstrata;
- "addEquipa" classe abstrata;
- "get_jogo_bilhetes" retorna a soma do número de adeptos das duas equipas que vão jogar;
- "toString" retorna uma String com o nome da liga e o número de espetadores total;

6.5 Torneio

Tem como objetivo criar um torneio que vai conter 4 competições selecionadas ou não pelo utilizador e depois de criada armazenar no ficheiro xml. É classe filha da classe Competicao o que torna possível o uso de "super()" que chama o método da classe pai. O construtor recebe o nome do torneio. Esta classe ainda está incompleta mas irá acabar de ser implementada na parte B do trabalho.

7 Conclusão

Apesar de o trabalho não estar terminado, creio que a aplicação concebida até agora está de acordo com o tipo de aplicação desejada pelos docentes, pois implementa os conhecimentos adquiridos ao longo de Modelação e Programação como a abstração e o armazenamento de dados em xml. Gostava de ter tido mais tempo para melhorar a aplicação e para acabar de implementar a classe Torneio o que vou fazer na parte B.