



## Modelação e Programação – TP4 Parte B



Docente: Eng João Ventura

# Introdução

O trabalho prático 4 tem como objetivo o desenvolvimento de uma aplicação usando a linguagem Java. A aplicação que eu decidi realizar é uma aplicação de futebol, esta aplicação tem como objetivo gerir uma liga de futebol, onde podemos adicionar e apagar equipas da liga, adicionar jogos, ver a classificação da liga, etc.

Para a realização do trabalho criei 6 classes, Comparador, Jogo, EquipaFutebol, Futebol, Ligae GUI\_Aplicacao.

Começando pela classe GUI\_Aplicacao, esta classe é a classe que gere toda a aplicação e trata da parte visual, a classe possui 4 métodos o build, o saveToXML, o main e o GUI\_Aplicacao.

O método build, é responsável por ler o arquivo XML, criar objetos do tipo Liga, EquipaFutebol e Jogo com base nos dados do arquivo e adicioná-los à instância do objeto Futebol.

O método saveToXML, é o inverso do método build, este ao invés de ler o que está no xml e criar objectos, quando chamado guarda os dados da aplicação no ficheiro XML.

Este arquivo xml contém várias informações como podemos ver nas imagens abaixo.

```
<futebol>
  <liga>
    <nome>plm</nome>
    <local>inglaterra</local>
    <numeroMax_clubes>7</numeroMax_clubes>
    <jogo>
      <teamA>United</teamA>
      <teamB>City</teamB>
      <teamAScore>2</teamAScore>
      <teamBScore>1</teamBScore>
      <date>2019-09-09</date>
    </jogo>
  </liga>
</futebol>
```

```
<equipa>
  <nome>Everton</nome>
  <local/>
</equipa>
</liga>
<liga>
  <nome>port</nome>
  <local>portugal</local>
  <numeroMax_clubes>0</numeroMax_clubes>
</liga>
<liga>
  <nome>esp</nome>
  <local>espanha</local>
  <numeroMax_clubes>0</numeroMax_clubes>
</liga>
</futebol>
```

FIGURA 1- EXEMPLO DO XML

O método main é o método principal que inicia a execução da aplicação. É criada uma instância da GUI\_Aplicacao e torna a GUI visível.

Depois o método GUI\_Aplicacao é o construtor da classe GUI\_Aplicacao que inicializa os componentes da interface gráfica. É aqui que toda a parte gráfica é manipulada, como vamos ver mais á frente nas funcionalidades da aplicação, dentro deste método são chamados outros métodos de outras classes que são essenciais para ligar a parte gráfica ao executável, como por exemplo getNomeEquipa() ou getGolosMarcados().

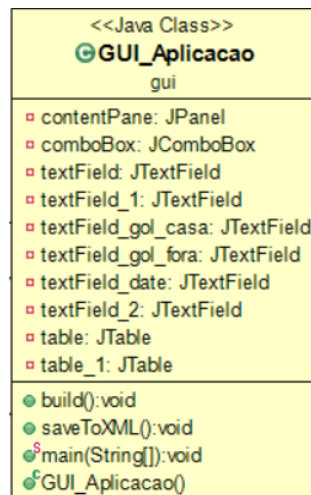


FIGURA 2- UML DA CLASSE GUI\_APLICACAO

A classe Jogo representa um jogo neste caso de futebol, onde tenho duas equipas neste caso team A e team B, e os seus golos também como atributos, vou ter ainda a data do jogo. Nos métodos tenho os getters e setters de cada atributo.

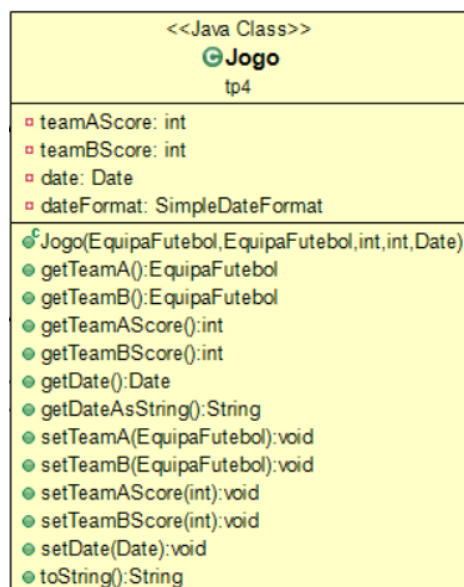


FIGURA 3- UML DA CLASSE JOGO

Na classe comparador implementamos a interface Comparador que tem como objetivo comparar duas equipas de futebol e colocá-las na tabela classificativa de acordo com o número de pontos.

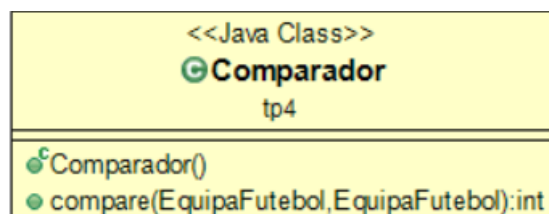


FIGURA 4- UML DA CLASSE COMPARADOR

A classe EquipaFutebol, esta classe tem atributos que parte de uma equipa como vitórias empates, derrotas, localizacao, etc, como podemos ver no diagrama uml. Depois tenho os getters e setters de cada atributo.

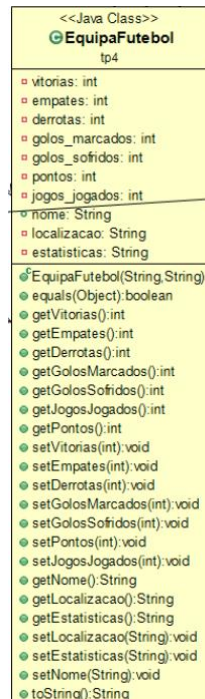


FIGURA 5- UML DA CLASSE EQUIPA FUTEBOL

A classe Liga, como o seu próprio nome indica representa uma Liga de futebol, a liga tem como atributos o nome da ligam, o local, o número máximo de clubes e o número atual de equipas na liga. Nos métodos tenho os setters e getters dos atributos, tenho o construtor que vai servir para criar objetos liga, depois tenho os métodos addEquipa e removeEquipa que servem para adicionar e remover equipas, tenho também os métodos para adicionar e remover jogos, e depois tenho também um método muito importante que é atualizarEstatisticas que recebe um jogo e vai atualizar as estatísticas das duas equipas presentes no jogo, o método é chamado no método build da classe GUI\_Aplicacao percorrendo os jogos presentes na liga usando o liga.getJogos, e atualizamos todas as estatísticas das equipas, também é chamado quando adicionamos um jogo para atualizar logo as estatísticas das equipas presentes nesse jogo.

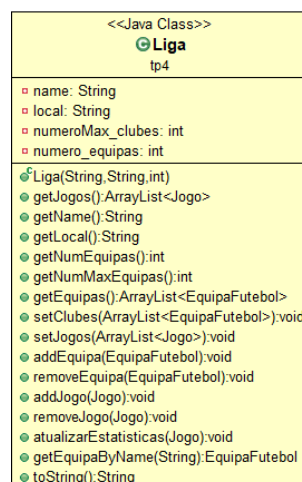


FIGURA 6- UML DA CLASSE LIGA

A classe Futebol, é responsável por gerir as ligas que a aplicação contém e adicioná-las ao array de ligas.

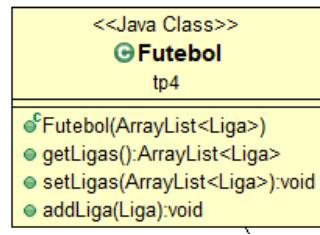


FIGURA 7- UML DA CLASSE FUTEBOL

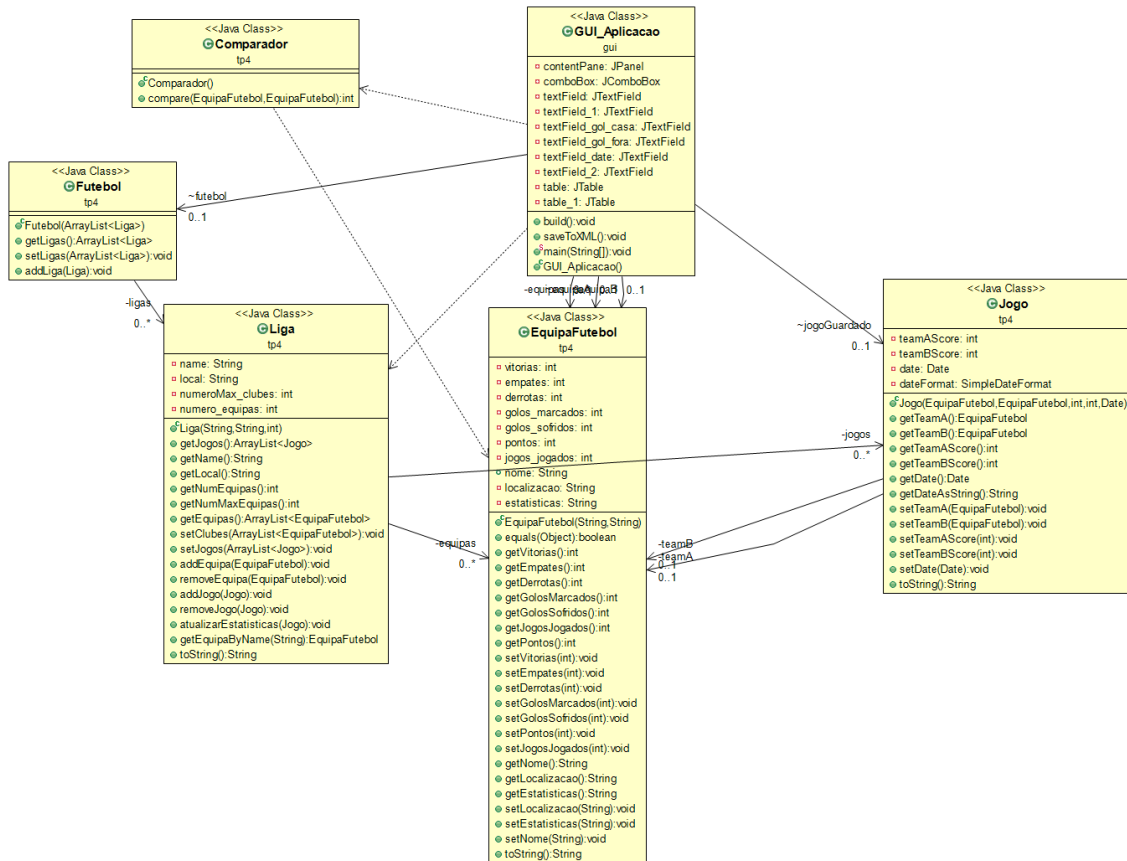
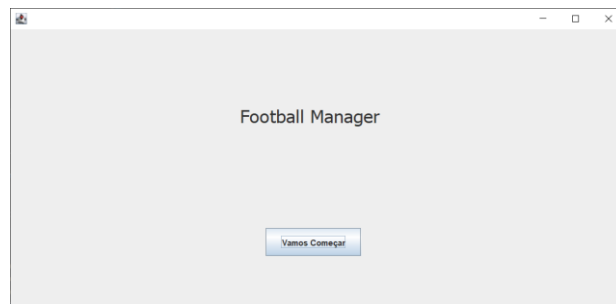


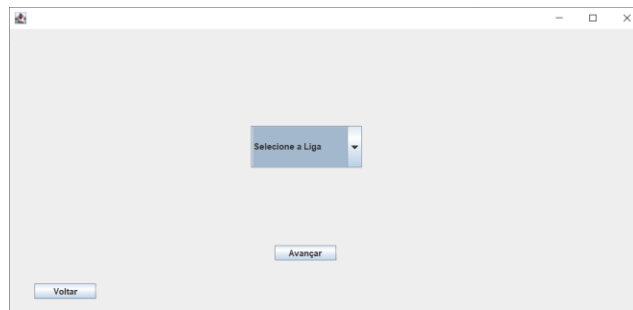
FIGURA 7- UML GERAL

## Parte Gráfica

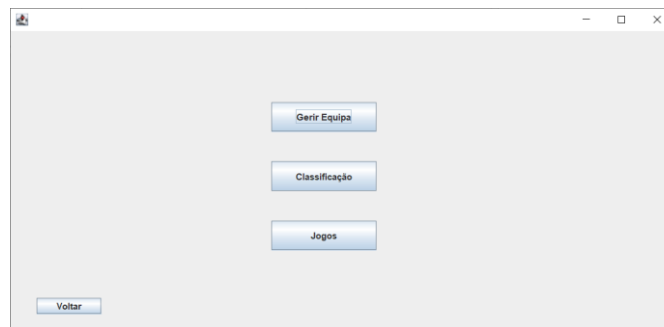
A aplicação começa com esta janela que serve como uma parte introdutória



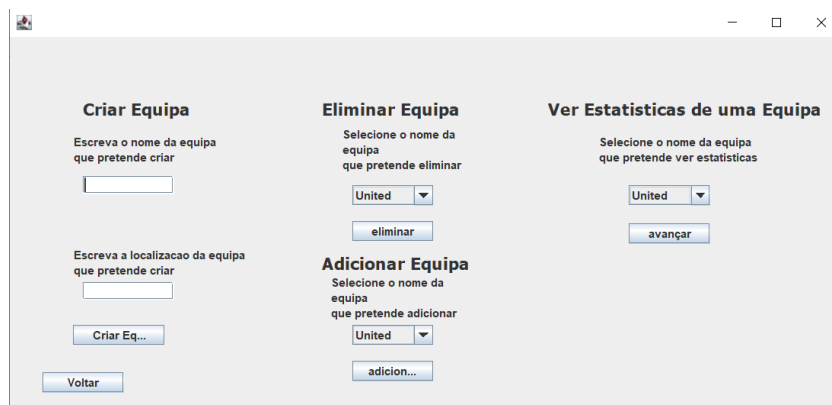
Na segunda janela podemos através de um JComboBox podemos seleccionar a liga que queremos trabalhar.



Na terceira janela podemos seleccionar o que pretendemos fazer se é gerir uma equipa, ver a classificação da liga ou entrar a parte dos jogos



Nesta janela podemos criar ou eliminar uma equipa e ainda ver as estatísticas de uma equipa.



Selecionando uma equipa por exemplo o United podemos observar nesta janela as suas respetivas estatísticas através dos getters presentes na classe EquipaFutebol.

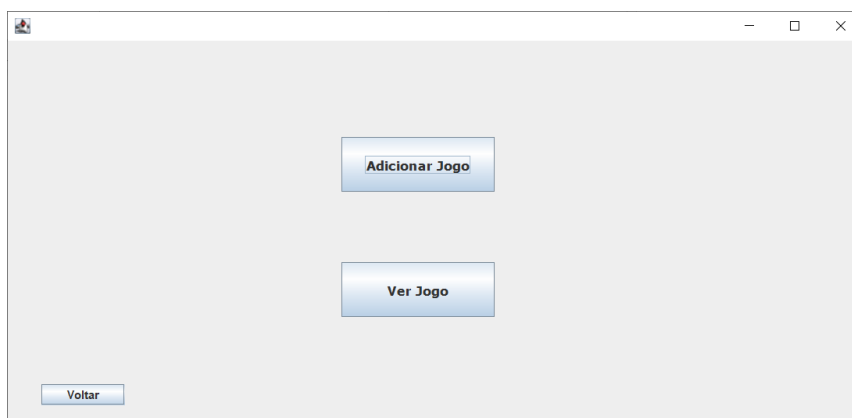


Clicando no botão classificação, podemos observar a classificação da liga que estamos a trabalhar, nesta liga a classe Comparador é essencial pois ordena as equipas de acordo com os seus pontos.

A janela de classificação da liga, com uma tabela de classificação e um botão "Voltar" no canto inferior direito.

Equipa	Posição	Pontos	Diferença de golos
United	1	4	1
City	2	3	2
Arsenal	3	3	1
Chelsea	4	1	0
Fulham	5	0	0
Liverpool	6	0	-1
Everton	7	0	-3

Clicando no botão jogos, vamos para esta janela onde podemos adicionar um jogo ou ver um jogo.

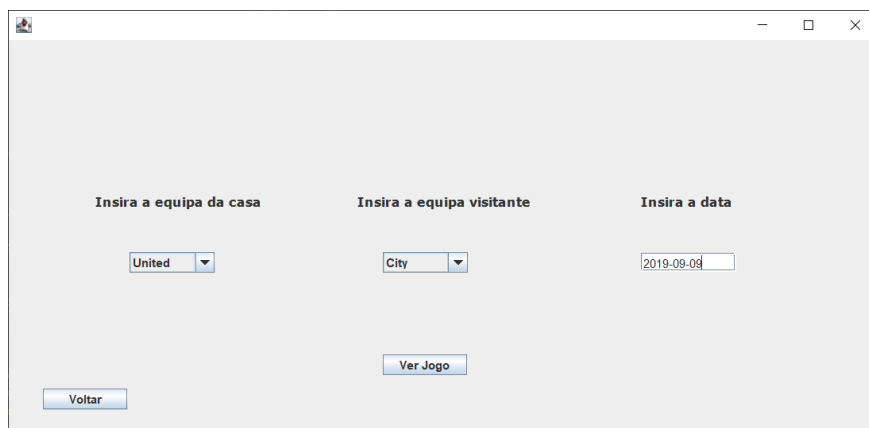


Clicando no botão adicionar jogo, podemos adicionar um jogo colocando as informações como equipa da casa, a equipa de fora e a data do jogo.



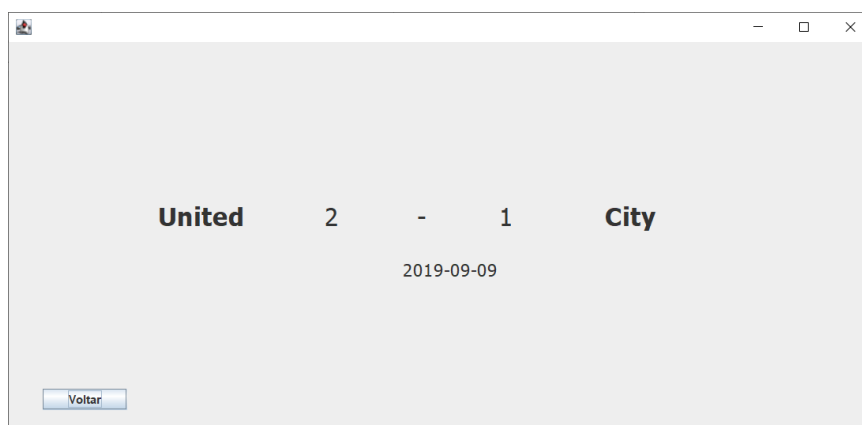
A screenshot of a software window titled "Adicionar Jogo". It contains four input fields: "Equipa da casa" (with a dropdown menu showing "United"), "Insira a data" (an empty text box), "Equipa Visitante" (with a dropdown menu showing "United"), and "Golos da equipa da casa" (an empty text box). There are also two buttons: "Adicionar Jogo" in the center and "Voltar" at the bottom left.

Clicando no botão ver jogo, vamos para esta janela onde colocamos as equipas do jogo e a data desse jogo.



A screenshot of a software window titled "Ver Jogo". It contains three input fields: "Insira a equipa da casa" (with a dropdown menu showing "United"), "Insira a equipa visitante" (with a dropdown menu showing "City"), and "Insira a data" (with a text box showing "2019-09-09"). There are two buttons: "Ver Jogo" in the center and "Voltar" at the bottom left.

Por fim, existindo o jogo nesta janela podemos ver as informações do jogo.



A screenshot of a software window titled "Ver Jogo" displaying the result of a game. The text "United 2 - 1 City" is centered, with "2019-09-09" below it. A "Voltar" button is at the bottom left.



## Conclusão

Para concluir, com este trabalho adquiri bastante conhecimento na área da programação orientada a objectos e também nas questões gráficas.

Também me enriqueceu muito na parte de gestão de dados, neste caso utilizando o xml, já que neste trabalho iniciando a aplicação os dados presentes no ficheiro xml vão logo ficar guardados, através do método build, não precisando de os criar cada vez que a aplicação é iniciada, com o saveToXML, guardamos no ficheiro xml o que adicionamos ou removemos na aplicação.