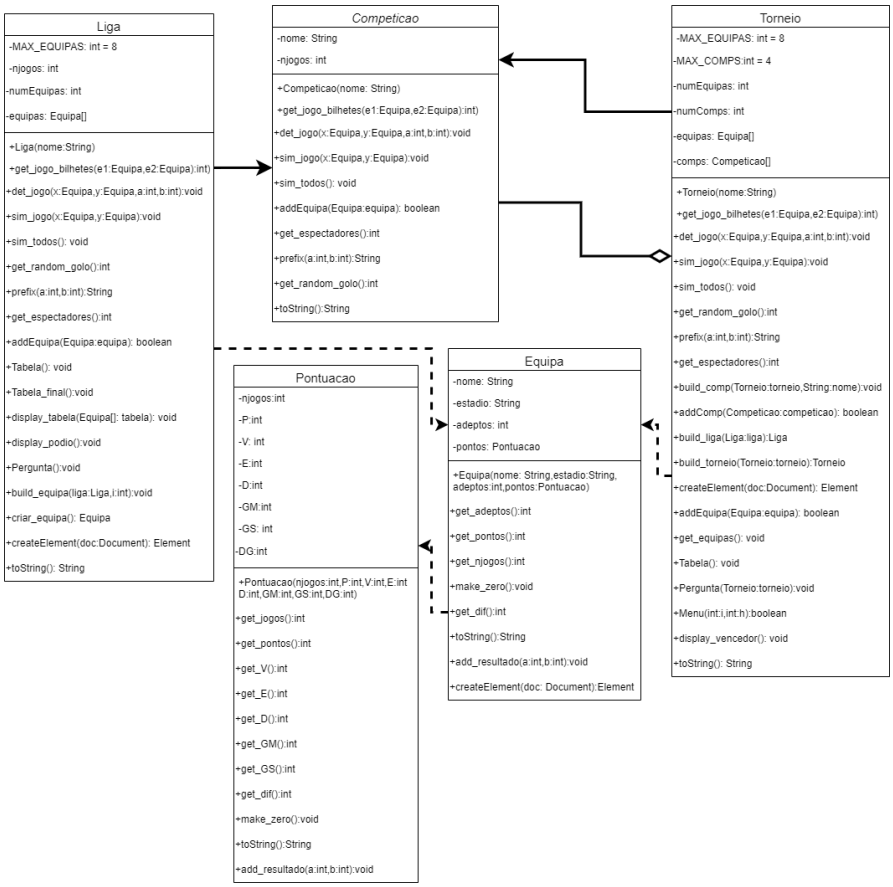
Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia

Modelação e Programação - MP - 2122SV

Trabalho Prático 4



Docente Carlos Júnior
Realizado por :
Pedro Silva 48965

26 de junho de 2022

Conteúdo

Lista de Figuras

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

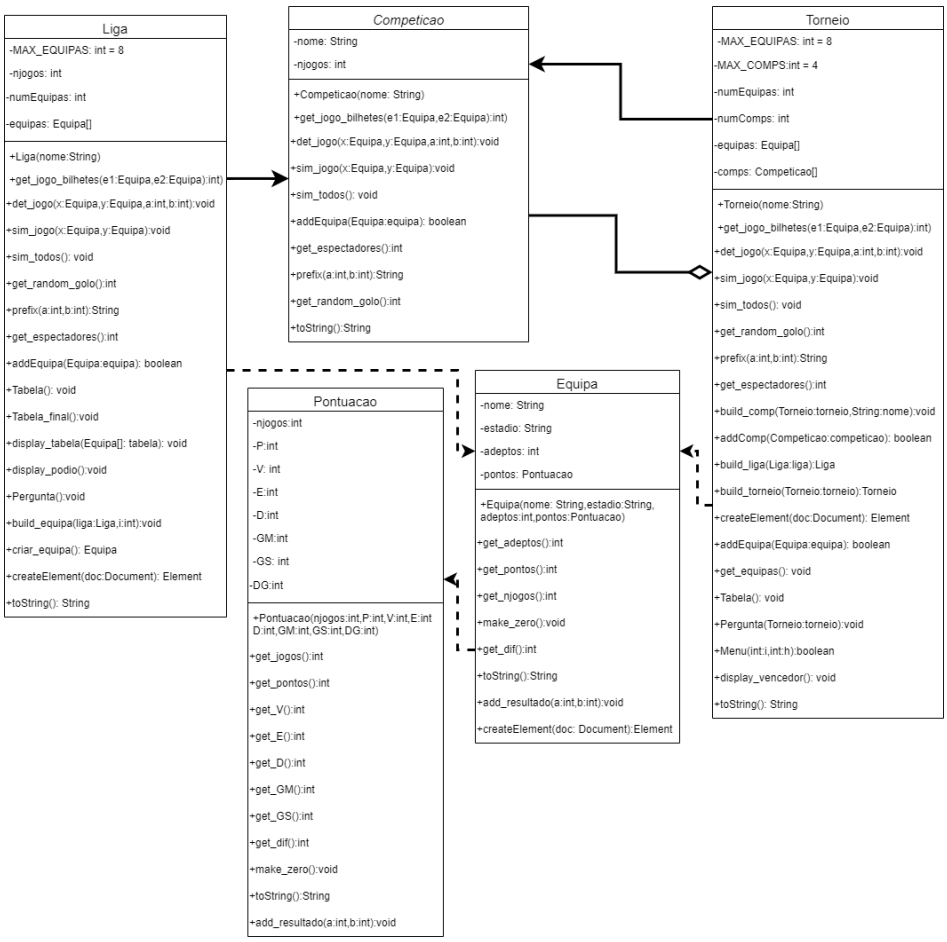


Figura 1: UML da aplicação Java

4 Wireframe da interface gráfica do simulador da Liga

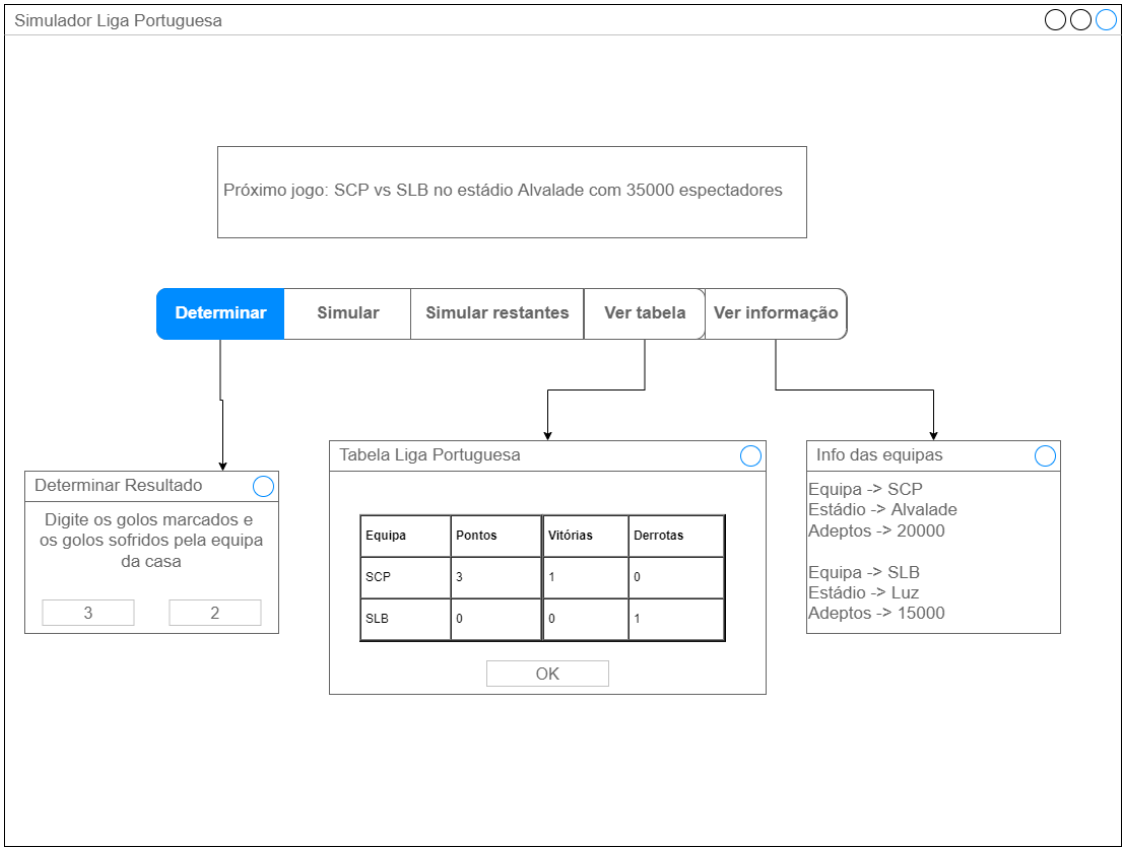


Figura 2: Wireframe simulador Liga

5 DTD

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

<!DOCTYPE BaseDados [
        <!ELEMENT BaseDados (Equipas|Ligas|Torneios)*>
        <!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>
        <!ELEMENT Torneios (Torneio)*>
        <!ELEMENT Torneio (Nome|Liga|Torneio)*>
        <!ATTLIST Torneio
                Comp CDATA #REQUIRED
                Vencedor 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 Main

Tem como objetivo criar todas as ligas e torneios auxiliares ao funcionamento do programa e também ligar os "mains"da classe liga e da classe torneio dando a escolha ao utilizador. Criamos 4 ligas e 2 torneios simulamos de forma a dar resultados sempre diferentes cada vez que iniciamos o programa e guardamos a informação no ficheiro ".xml".

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 sendo a probabilidade de esse número maior quanto maior o valor e vice-versa;
- "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;
- "createElement"classe abstrata;

6.5 Liga

Tem como objetivo criar uma liga que vai conter 8 equipas 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 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"organiza a tabela classificativa ao comparar o numero de pontos de cada equipa.No caso de uma equipa ter os mesmos pontos que outra verificamos a difernça de golos;

- "Tabela_final"imprime na consola a tabela classificativa da liga e organiza o array da liga de acordo com a classificação de forma a este estar pronto a ser guardado no ficheiro ".xml";
- "Pergunta"pergunta ao utilizador se quer criar novas equipas para adicionar à liga.Se sim chamamos "criar_equipa"se não o programa vai ao ficheiro ".xml"buscar dentro de "Equipas"equipas para adicionar a liga até chegar a 8 equipas;
- "build_equipa"adiciona à liga equipas previamente criadas a partir de um ficheiro xml utilizando o XPath. Devolve true se a equipa foi adicionada com sucesso;
- "criar_equipa"Retorna uma equipa criada pelo utilizador através de "JOptionPane"que criam uma janela que pede ao utilizador as várias informações da equipa;
- "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. O programa percorre o número de equipas da liga e adiciona ao elemento "eLiga"através do método na classe Equipa com o mesmo nome;
- "main"cria, adiciona, simula e guarda no ficheiro ".xml"no elemento Equipas uma equipa auxiliar que vai ser utilizada no caso de o utilizador não criar todas as equipas necessárias para criar a liga. Cria um menu que vai perguntar ao utilizador para cada jogo da liga se quer: determinar o resultado, simular o jogo, simular os jogos restantes, ver a tabela, ver a informação de cada equipa ou sair do programa. Para percorrer os jogos criamos dois loops em que de cada vez uma equipa vai fazer 7 jogos seguidos contra as restantes equipas. Utiliza-se um "switch"para cada uma das opções, depois de todos os jogos serem realizados exibe-se a tabela e guarda-se no ".xml".

6.6 Torneio

Tem como objetivo criar um torneio que vai conter 4 competições , duas delas ligas obrigatoriamente, 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. Os métodos desta classe são:

- "get_jogo_bilhetes"chama a superclasse com o mesmo nome;
- "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". Enquanto o número de equipas for maior que 2 simulamos os jogos do torneio e eliminamos as equipas com menor diferença de golos nos dois jogos. No inicio de cada fase tornamos os golos das equipas em zero. Para cada mão verificamos se as equipas já realizaram o jogo respetivo se sim passamos para o próximo par de equipas. No final de cada fase colocamos a null a equipa com maior

diferença de golos e eliminamos-a do array de equipas. Quando só faltarem 2 equipas voltamos a colocar a zero a diferença de golos das equipas simulamos o jogo escrevemos o vencedor na consola e eliminamos a equipa perdedora;

- "prefix"chama a superclasse com o mesmo nome;
- "get_espectadores"devolve a soma do número de espetadores de todas as equipas do torneio;
- "build_comp"Constrói uma nova competição a partir de um ficheiro ".xml"e adiciona ao torneio recebido. Recebemos uma string com o nome da competição que vai ser adicionada ao torneio. Através do xpath e do nome da competição procuramos o atributo "comp"da competição, se for "L"criamos uma liga se for "T"criamos um torneio usando o método "add_comp";
- "addComp"adiciona a competição recebida ao torneio. Se a competição recebida for null ou não existir mais espaço retorna "false". Percorremos as competições e se tiverem o mesmo nome que a que pretendemos criar lançamos uma exceção a alertar o utilizador. Se a competição for uma instância de Liga construimos a liga com o "build_liga"e adicionamos à lista de competições, se não, é instância de Torneio logo construimos o torneio com o "build_torneio", fazendo primeiro um cast da competição para o tipo Torneio e adicionamos à lista de competições;
- "build_liga"Constrói a liga com os dados da liga com o mesmo nome no ficheiro ".xml"e devolve-a preenchida. Recebemos uma Liga com o nome de uma liga já existente e usamos esse nome para procurar no ficheiro a sua informação. Percorremos o número de equipas da liga, criamos-as, adicionamos-as à liga recebida e devolvemos essa liga;
- "build_torneio"Constrói o torneio com os dados do torneio com o mesmo nome no ficheiro ".xml"e devolve-a preenchida. Recebemos um Torneio com o nome de um torneio já existente e usamos esse nome para procurar no ficheiro a sua informação. Começamos por contar o número de ligas e o número de torneios dentro do torneio e guardamos o nome de cada competição. Para o número de ligas criamos uma liga com o nome guardado anteriormente e adicionamos ao torneio que recebemos através do método "addComp", o mesmo acontece com os torneios. Depois criamos a única equipa no torneio, ou seja, o vencedor ao guardarmos numa variável o nome da equipa que está no atributo vencedor, procuramos em todas equipas com o nome guardado o estádio e o número de adeptos,utilizando o xpath. Por fim adicionamos essa equipa ao novo torneio e retornamos-o;
- "createElement"Devolve um elemento "eTorneio"que vai ser guardado posteriormente no ficheiro ".xml"dentro do elemento "Torneios".Percorre as competições cria o seu elemento e adiciona ao elemento "eTorneio".Adiciona o atributo vencedor com a única equipa existente no torneio;
- "addEquipa"Adiciona a equipa recebida à lista de equipas participantes neste torneio;
- "get_equipas"Adiciona à lista de equipas deste torneio as equipas a partir das competições presentes neste torneio. Começamos por contar o número de ligas existentes no torneio e dependendo do número, o número de equipas adicionadas por liga varia, mas por torneio mantém-se, sendo apenas o vencedor

adicionado. Se existirem 2 ligas no torneio quer dizer que tem também dois torneios logo precisamos de 3 equipas de cada liga e do vencedor de cada torneio. Se existirem 3 ligas no torneio quer dizer que tem um torneio logo precisamos de 3 equipas de duas ligas, 1 de uma liga e do vencedor do torneio. Se só existirem ligas no torneio quer dizer precisamos de 2 equipas de cada liga. Se não existir pelo menos 2 ligas lança exceção para alertar o utilizador que o número mínimo de ligas não foi atingido;

- "Tabela"exibe numa janela os jogos a ser realizados na fase da competição. Usamos a imagem "bracket_8.png"como uma label, criamos labels para todas as equipas do torneio e posicionamos segundo a imagem. Utilizamos um "option pane"para ser necessário o input do utilizador para prosseguir.

- "Pergunta"Pergunta ao utilizador se quer adicionar as competições manualmente ou se quer que isso seja feito automaticamente. Se sim percorremos o número de competições do torneio e incrementamos a variável auxiliar "count"se encontrarmos uma liga, se esta for maior ou igual a 2 informamos o utilizador que o número mínimo de ligas foi adicionado. Perguntamos se quer adicionar uma liga ou um torneio, se quiser adicionar uma liga guardamos num array os nomes das ligas disponíveis (ao utilizar o xpath e percorrer o ficheiro xml) para o utilizador escolher. O utilizador efetua a sua escolha através de uma janela com uma "dropbox", a liga escolhida vai ser construida e adicionada ao torneio através do "build_comp". O mesmo ocorre se for escolhido um torneio. Se o utilizador não inserir o número máximo de competições inserir automaticamente as competições restantes a partir do ficheiro xml, mais especificamente as ligas.

- "Menu"Menu em que o utilizador pode escolher determinar ou simular um jogo, simular os restantes, ver o estado da fase do torneio ou a informação das equipas que vão jogar.Retorna um booleano "true"se for selecionada a opção de simular os jogos restantes o que nos vai ajudar a saltar passos que seguem. Criamos uma variável "again"que vai nos ajudar a repetir o loop, pois quando as opções da tabela e da informação são selecionadas o jogo não é realizado logo queremos repetir o ciclo. Utiliza-se um "JOptionPane"com botões para cada uma das opções possíveis.

- "Main"Cria um novo torneio com o nome dado pelo utilizador e pergunta como quer adicionar as competições chamando o método "Pergunta". Adiciona as equipas ao torneio("get_equipas") e exibe as equipas participantes no torneio("Tabela"). Utilizamos o mesmo método que foi utilizado em "sim_todos"para percorrer cada jogo e vamos chamando "Menu"para dar a escolher ao utilizador o que fazer em cada jogo, se for retornado "true"saímos do loop pois todos os jogos foram realizados. Para a final escolhemos uma das capitais da Europa para se realizar o último encontro. Para a final o utilizador tem as opções determinar o resultado, simular ou ver a informação das equipas. Por fim criamos o elemento do Torneio e guardamos no ficheiro ".xml".

7 Interface

Esta parte do trabalho tem como objetivo criar uma interface gráfica para o programa, que vai substituir todas as interações com o utilizador que ocorriam na consola. Para que isso aconteça foram implementados:

- No main uma janela que dá as boas-vindas ao utilizador e pergunta se quer simular uma liga ou um torneio sendo essa resposta captada por dois botões.

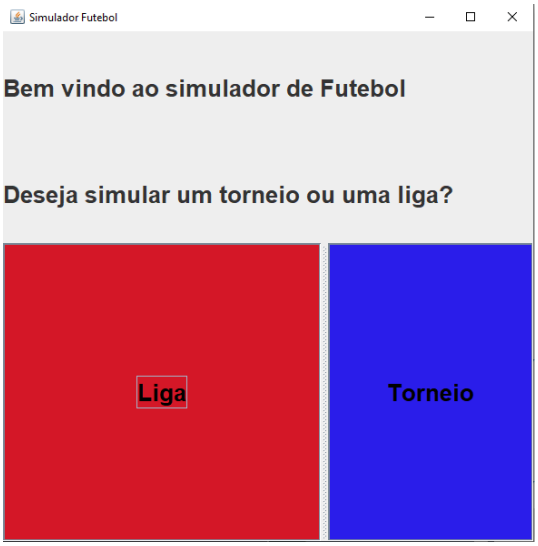


Figura 4: Main

- A substituição de todos os pedidos de informação na consola por janelas através "JOptionPane.showInputDialog"

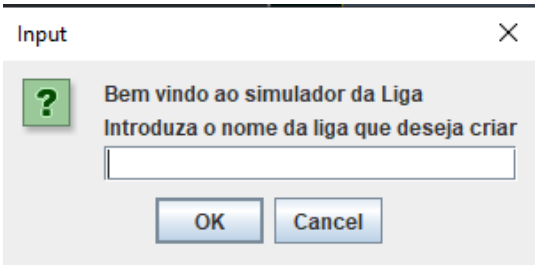


Figura 5: JOptionPane.showInputDialog

- A substituição de todos as escolhas na consola por janelas através "JOptionPane.showConfirmDialog"e "JOptionPane.showOptionDialog"

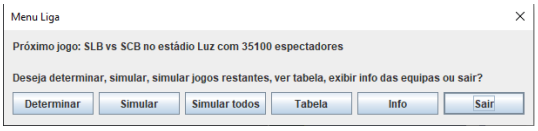


Figura 6: JOptionPane.showConfirmDialog

- A implementação de uma tabela numa janela em vez de na consola ao criando o método "display_tabela"que guarda numa "JTable"toda a informação à cerca da pontuação.

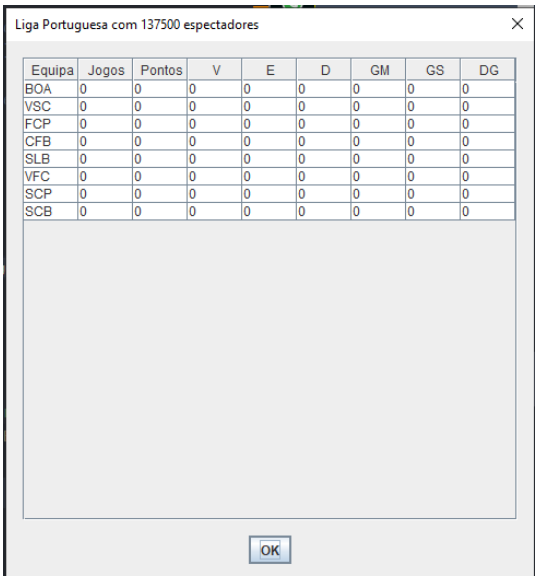


Figura 7: Tabela Liga

- A criação de um pódio que exibe o top 3 da liga pelo método "display_podio"que usa uma imagem de um pódio e coloca sobre ela os nomes das equipas .

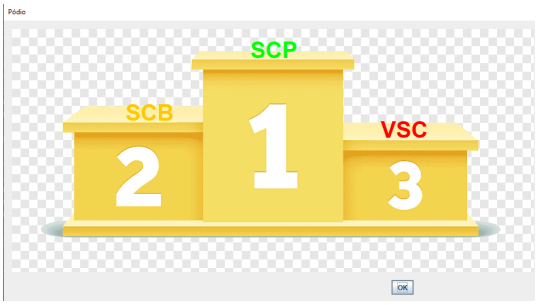


Figura 8: Pódio

- A criação de uma tabela que exibe os jogos que vão ser realizados na fase do torneio pelo método da classe Torneio "Tabela"que usa uma imagem de uma "bracket"e coloca sobre ela os nomes das equipas.

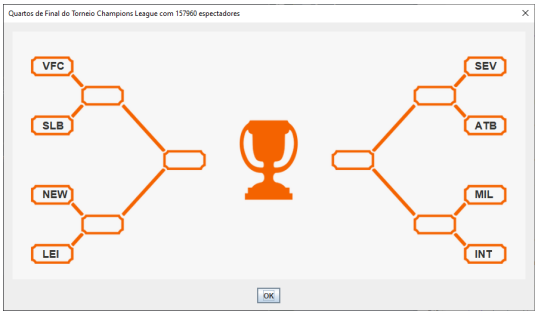


Figura 9: Quartos de Final

- A criação de uma imagem que exibe o vencedor do torneio ao criar o método "display_vencdeor"que usa uma imagem de uma troféu e coloca sobre ela o nome da equipa vencedora.

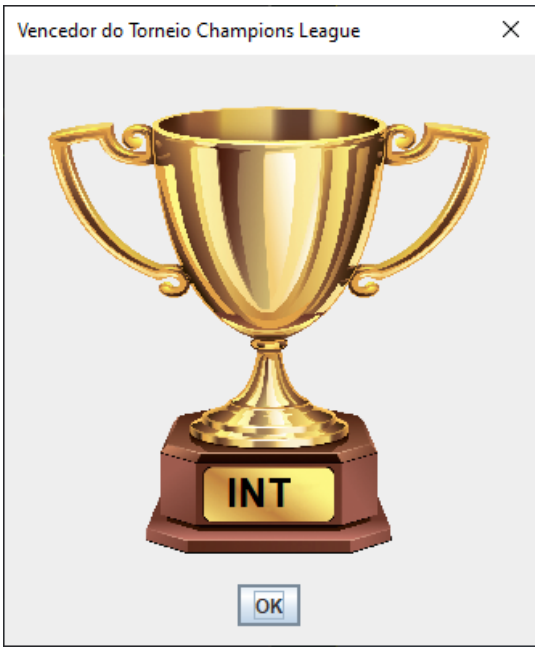


Figura 10: Troféu

8 Conclusão

O objetivo deste trabalho era o de projetar um projeto Java que implementasse todos os conhecimentos adquiridos ao longo de Modelação e Programação como a abstração, o armazenamento de dados em xml e a interface gráfica. Ao desenvolver este projeto consolidei a matéria aprendida e também adquiri novas formas de acrescentar riqueza ao trabalho e à interface gráfica. Creio que a aplicação concebida até agora está de acordo com o tipo de aplicação desejada pelos docentes, tanto na complexidade e originalidade, como demonstra o domínio que tenho sobre a matéria lecionada. O único problema do programa é o de ao adicionar informação ao ficheiro xml cria espaços entre a informação previamente lá guardada, pois tirando isso acredito que consegui realizar tudo o que tinha projetado.