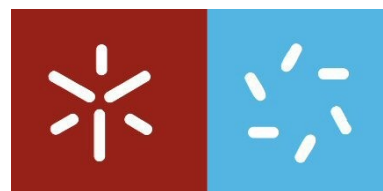


**UNIVERSIDADE DO MINHO**  
**LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO**



**PROGRAMAÇÃO ORIENTADA AOS OBJETOS**  
**TRABALHO PRÁTICO**  
**GRUPO 15**

**José Pedro Gomes  
Ferreira**



**A91636**

**Rui Jordão Sampaio  
Gonçalves**



**A91652**

**Tiago André Oliveira Leite**



**A91693**

**Ano Letivo 2020/2021**

## Índice

1. Introdução.....	4
2. Classes.....	5
2.1 Main.....	5
2.2 State.....	5
2.3 FootballTeam.....	5
2.4 FootballPlayer.....	6
2.4.1 Goalkeeper.....	7
2.4.2 Defender.....	7
2.4.3 Winger.....	7
2.4.4 MildFielder.....	8
2.4.5 Striker.....	8
2.5 FootballMatch.....	9
2.6 Controller.....	10
2.7 MatchController.....	10
2.8 Menu.....	11
2.9 IO.....	11
2.10 Excepções.....	11
2.10.1 WrongLineException.....	11
2.10.2 NoTeamsException.....	11
2.10.3 SubstitutionsException.....	11
3. Enums.....	12
3.1 Position.....	12
3.2 MatchState.....	12
4. Arquitetura de Camadas.....	13
5. Implementação das Funcionalidades.....	14
5.1 Menu Principal.....	14
5.2 Menu Jogo.....	14
5.2.1 Escolha Equipa.....	15
5.2.2 Titulares.....	15
5.2.3 Banco.....	16
5.2.4 Substituição.....	16
5.2.5 Substituição Programada.....	17
5.2.5 Escolha Tática.....	18
5.2.6 Simulação do Jogo.....	18
5.2.7 Cálculo do Resultado.....	19
5.3 Menu Jogos Realizados.....	19
5.4 Menu Jogadores.....	20
5.4.1 Consultar Jogador.....	20
5.4.2 Transferir Jogador.....	20
5.4.3 Criar Jogador.....	21
5.4.4 Apagar Jogador.....	21
5.5 Menu Equipas.....	22
5.6.1 Consultar Equipa.....	22
5.6.2 Criar Equipa.....	22

5.6.3 Apagar Equipa.....	23
5.7 Guardar Estado.....	23
5.8 Carregar Estado.....	23
6. Diagrama de Classes.....	24
7. Conclusão.....	25

## **1. Introdução**

Com este projeto foi nos proposto o desenvolvimento de um programa em Java que funcione como um sistema de gestão e simulação de equipas e jogos de futebol, um pouco à semelhança do conhecido jogo Football Manager.

No desenvolvimento do programa procuramos sempre respeitar os princípios da programação orientada a objetos, esperando por isso que o resultado final cumpra esses requisitos.

Consideramos que o principal desafio deste projeto foi a simulação de um jogo de futebol, pois a interação com o utilizador, torna necessário pensar em todos os cenários e inputs possíveis que o nosso programa pode receber.

## 2. Classes

### 2.1 Main

Classe da camada de lógica que é responsável por iniciar o programa. Tem como função instanciar um objeto da classe state, tentando de seguida carregar os dados das equipas, jogadores e jogos do ficheiro de logs, fornecido pelos docentes da UC, para o objeto state. De seguida instancia um objeto da classe controller com o objeto state passado como parâmetro e coloca o controller a correr.

### 2.2 State

Classe da camada de dados responsável por armazenar toda a informação dos jogadores, equipas e jogos realizados.

### 2.3 FootballTeam

```
private String name;  
private Map<Integer, FootballPlayer> squad;
```

Classe da camada de dados que define as equipas de futebol. Contém a informação do nome da equipa (**name**) e um map de jogadores (**squad**) cuja chave é o número de camisola do jogador.

Entre outros, contem o metodo **public double overall** que permite calcular a habilidade da equipa (plantel). Este cálculo é feito somando a habilidade de todos os jogadores e dividindo pelo número total de jogadores do plantel.

## 2.4 FootballPlayer

```
private String name;  
private int number;  
private int speed;  
private int stamina;  
private int agility;  
private int jumping;  
private int heading;  
private int finishing;  
private int passing;  
private String team;  
private ArrayList<String> career;
```

Classe abstrata, da camada de dados, que define o jogador de futebol. Todos os tipos de jogadores herdam as seguintes características:

- Nome (**name**);
- Número da camisola (**number**);
- Velocidade (**speed**);
- Resistencia (**stamina**);
- Destreza (**agility**);
- Impulsão(**jumping**);
- Jogo de Cabeça (**heading**);
- Remate (**finishing**);
- Capacidade de passe(**passing**);
- Equipa atual (**team**);
- Clubes anteriores (**carrer**);

Entre outros, contem o metodo **public abstract int overall** que permite calcular a habilidade de um jogador.

### 2.4.1 Goalkeeper

```
private int elasticity;
```

Subclasse da classe **FootballPlayer**, que define o tipo de jogador Guarda-Redes.

Os guarda-redes para além dos atributos herdados da classe **FootballPlayer**, possuem também o atributo elasticidade (**elasticity**).

O cálculo da habilidade é feito através da formula:

$(\text{velocidade} \times 0,8 + \text{resistencia} \times 0,8 + \text{destreza} \times 1,8 + \text{impulsão} \times 1,7 + \text{jogo de cabeça} \times 0,2 + \text{remate} \times 0,2 + \text{capacidade de passe} \times 0,5 + \text{elasticidade} \times 0,2) / 8$ .

### 2.4.2 Defender

```
private int marking;
```

Subclasse da classe **FootballPlayer**, que define o tipo de jogador Defesa, que no nosso trabalho é o jogador que joga no centro da defesa.

Os defesas para além dos atributos herdados da classe **FootballPlayer**, possuem também o atributo marcação (**marking**).

O cálculo da habilidade é feito através da formula:

$(\text{velocidade} \times 0,8 + \text{resistencia} + \text{destreza} \times 0,8 + \text{impulsão} \times 1,5 + \text{jogo de Cabeça} \times 1,2 + \text{remate} \times 0,4 + \text{capacidade de passe} \times 0,8 + \text{marcação} \times 1,5) / 8$ .

### 2.4.3 Winger

```
private int crossing;
```

Subclasse da classe **FootballPlayer**, que define o tipo de jogador Lateral, que no nosso trabalho é o jogador que joga como defesa lateral.

Os laterais para além dos atributos herdados da classe **FootballPlayer**, possuem também o atributo cruzamento (**marking**).

O cálculo da habilidade é feito através da formula:  
(velocidade x 1,5 + resistencia x 1,5 + destreza x 1,5 + impulsão x 0,2 + jogo de cabeça x 0,2 + remate x 0,8 + capacidade de passe x 0,8 + cruzamento x 1,5 ) / 8.

#### 2.4.4 MildFilder

```
private int recovery;
```

Subclasse da classe **FootballPlayer**, que define o tipo de jogador Médio, que nosso trabalho é o jogador que joga no meio campo.

Os médios para além dos atributos herdados da classe **FootballPlayer**, possuem também o atributo recuperação (**recovery**).

O cálculo da habilidade é feito através da formula:  
(velocidade x 0,8 + resistencia x 1,5 + destreza + impulsão x 0,5 + jogo de cabeça x 0,4 + remate x 0,8 + capacidade de passe x 1,5 + recuperação x 1,5 ) / 8.

#### 2.4.5 Striker

```
private int positioning;
```

Subclasse da classe **FootballPlayer**, que define o tipo de jogador Avançado, que nosso trabalho é o jogador que joga no ataque.

Os avançados para além dos atributos herdados da classe **FootballPlayer**, possuem também o atributo posicionamento (**positioning**).

O cálculo da habilidade é feito através da formula:  
(velocidade + resistencia x 0,8 + destreza + impulsão + jogo de cabeça + remate x 1,2 + capacidade de passe x 0,8 + posicionamento x 1,2 ) / 8.



## 2.5 FootballMatch

```
public static final int [][] tatics = { {1,2,2,4,2}, {1,2,2,3,3}, {1,3,2,3,2}};  
private String teamHome;  
private String teamAway;  
private int scoreHome;  
private int scoreAway;  
private LocalDate date;  
private MatchState state;  
private Map<Integer, FootballPlayer> squadHome;  
private Map<Integer, FootballPlayer> squadAway;  
private List<Integer> playersHome;  
private List<Integer> playersAway;  
private Map<Integer, Integer> substitutionsHome;  
private Map<Integer, Integer> substitutionsAway;  
private int[] taticHome;  
private int[] taticAway;
```

Classe da camada de dados que define o jogo de futebol. Tem como variáveis de instância o nome das equipas (**teamHome** e **teamAway**), o número de golos marcados por cada equipa durante o jogo (**scoreHome** e **scoreAway**), a data do jogo(**date**), o momento atual do jogo(**state**), isto é, se ainda não começou, se estamos na primeira parte, intervalo, segunda parte ou se já acabou. Contem também dois Maps onde estão armazenados os plantéis (**squadHome** e **squadAway**) cuja chave é o número de camisola, duas Lists com o número de camisola dos jogadores que estão em campo (**playersHome** e **playerAway**), dois Maps onde se armazenam as substituições realizadas por cada equipa (**substitutionsHome** e **substitutionsAway**) e as táticas escolhidas por cada equipa(**taticHome** e **taticAway**).

Entre outros, contem os metodos:

- **public double overallHome** – calcula a habilidade da equipa da casa titular em função da tática e jogadores escolhidos;
- **public double overallAway** – calcula a habilidade da equipa visitante titular em função da tática e jogadores escolhidos;
- **public double defensiveOverallHome** - calcula a habilidade defensiva da equipa da casa titular em função da tática e jogadores nas posições defensivas;
- **public double defensiveOverallAway** - calcula a habilidade defensiva da equipa visitante titular em função da tática e jogadores nas posições defensivas;

- **public double offensiveOverallHome** - calcula a habilidade ofensiva da equipa da casa titular em função da tática e jogadores nas posições ofensivas;
- **public double offensiveOverallAway** - calcula a habilidade ofensiva da equipa visitante titular em função da tática e jogadores nas posições ofensivas;

## 2.6 Controller

```
State state;
```

Classe da camada de lógica responsável por gerir o fluxo da aplicação. Faz a ligação entre a camada de dados e a camada de interface com o utilizador. Tem um objeto do tipo State (**state**) como variável de instância.

## 2.7 MatchController

```
private FootballMatch match;  
private int attacking;  
private Map<Integer,Map<Integer, Integer>> futureSubstitutionsHome;  
private Map<Integer,Map<Integer, Integer>> futureSubstitutionsAway;
```

Classe da camada de lógica responsável por controlar a uma partida de futebol, fazendo a ligação entre a camada de dados e a camada de interface com o utilizador. Tem como variáveis de instância um jogo de futebol (**match**) um valor do tipo inteiro (**attacking**) que identifica a equipa que está a atacar e também dois Maps (**futureSubstitutionsHome**, **futureSubstitutionsAway**) que armazenam as substituições programadas pelo utilizador assim o momento em que estas devem ocorrer, caso o utilizador opte por programar substituições.

Esta classe é responsável por calcular o resultado de um jogo de futebol durante os vários momentos da partida, invocando para tal métodos do objeto match que lhe permitem saber, em cada momento do jogo, qual o poder ofensivo e defensivo de cada uma das equipas.

## 2.8 Menu

```
private List<String> options;  
private int option;  
private String title;  
private String exitOption;
```

Classe da camada de interface responsável pela criação dos menus para interação com o utilizador. Tem como variáveis de instância uma List de opções (**options**) para o utilizador escolher, um valor do tipo inteiro que representa a escolha do utilizador (**option**), o título de cada menu (**title**) e a opção para voltar atrás(**exitOption**).

## 2.9 IO

Classe da camada de interface composta por métodos estáticos que tem como responsabilidade interagir com o utilizador. Ou seja, obter informação e transmitir informação.

## 2.10 Excepções

### 2.10.1 WrongLineException

Exceção que pode ocorrer durante a leitura do ficheiro fornecido pelos docentes da UC.

### 2.10.2 NoTeamsException

Exceção que ocorre caso o utilizador tente realizar um jogo de futebol mas não exista nenhuma equipa com o número suficiente de jogadores.

### 2.10.3 SubstitutionsException

Exceção que ocorre quando o utilizador faz uma substituição inválida ou tenta realizar mais substituições do que as permitidas.

### 3. Enums

#### 3.1 Position

GOALKEEPER DEFENDER WINGER MIDFIELDER STRIKER
---

Tipo de dado enumerável que identifica qual a posição de campo do jogador de futebol.

#### 3.2 MatchState

TOSTART FIRSTHALF SECONDDHALF INTERVALL FINISHED
--

Tipo de dado enumerável que identifica qual o momento da partida de futebol.

## 4. Arquitetura de Camadas

O nosso trabalho utiliza uma divisão por camadas que tenta seguir a estrutura model-view-controller.

O **model** (camada de dados) é composto pelas classes:

- State;
- FootballPlayer e as suas subclasses;
- FootballTeam;
- FootballMatch.

O **controller** (camada de lógica) é composto pelas classes:

- Controller;
- MatchController.

A **view** (camada de interface) é constituída pelas classes:

- Menu;
- IO.

As classes da camada de dados, são responsáveis pelo tratamento e armazenamento da informação da aplicação enquanto que as classes da camada de interface são responsáveis por interagir com o utilizador. As classes da camada de lógica funcionam como uma ponte entre a camada de dados e a camada de interface. Por exemplo, fazem um pedido à camada de interface, analisam a informação recebida e invocam o método que acharem necessário da camada de dados.

## 5. Implementação das Funcionalidades

### 5.1 Menu Principal

```
*** Menu ***  
1. Fazer Jogo  
2. Resultados  
3. Jogadores  
4. Equipas  
5. Guardar Estado  
6. Carregar Estado  
0. Sair
```

### 5.2 Menu Jogo

```
*** Tipo de Jogo ***  
1. Jogo Completo  
2. Simular Resultado  
0. Sair  
Opção:
```

## 5.2.1 Escolha Equipa

```

*** Escolha Equipa ***
1. Bach F. C.
2. Bartok F. C.
3. Beethoven F. C.
4. Brahms F. C.
5. Debussy Athletic
6. Handel Athletic
7. Mahler Athletic
8. Mendelssohn F. C.
9. Mozart F. C.
10. Schumann Athletic
11. Sporting Club Chopin
12. Sporting Club Dvorak
13. Sporting Club Prokofiev
14. Sporting Club Schubert
15. Sporting Club Shostakovich
16. Stravinsky Athletic
17. Vivaldi F. C.
18. Wagner Athletic
0. Sair
Opção:

```

## 5.2.2 Titulares

```

*** Titulares ***
30 | Rafael Diogo Batista Carvalho | Guarda-Redes
Overall: 64 | Velocidade: 67 | Resistencia: 52 | Destreza: 51 | Impulsao: 81 | Jogo de Cabeça: 14 | Remate: 67 | Capacidade de Passe: 68 | Elasticidade: 70
34 | Afonso Brandao Cruz | Defesa
Overall: 60 | Velocidade: 42 | Resistencia: 82 | Destreza: 50 | Impulsao: 58 | Jogo de Cabeça: 71 | Remate: 57 | Capacidade de Passe: 75 | Marcação: 50
24 | Lucas da Silva Carvalho | Defesa
Overall: 57 | Velocidade: 53 | Resistencia: 72 | Destreza: 19 | Impulsao: 52 | Jogo de Cabeça: 96 | Remate: 21 | Capacidade de Passe: 56 | Marcação: 50
36 | Diogo Francisco Lima Barros | Lateral
Overall: 71 | Velocidade: 70 | Resistencia: 70 | Destreza: 81 | Impulsao: 33 | Jogo de Cabeça: 9 | Remate: 83 | Capacidade de Passe: 72 | Cruzamento: 73
47 | Goncalo Joao de Oliveira Barbosa Lobarinhas Henriques | Lateral
Overall: 66 | Velocidade: 98 | Resistencia: 92 | Destreza: 54 | Impulsao: 90 | Jogo de Cabeça: 56 | Remate: 75 | Capacidade de Passe: 73 | Cruzamento: 14
5 | Rodrigo Caldas Meira | Medio
Overall: 74 | Velocidade: 96 | Resistencia: 53 | Destreza: 91 | Impulsao: 84 | Jogo de Cabeça: 63 | Remate: 78 | Capacidade de Passe: 94 | Recuperacao: 54
43 | Andre Pizarro Martins | Medio
Overall: 66 | Velocidade: 63 | Resistencia: 57 | Destreza: 56 | Impulsao: 53 | Jogo de Cabeça: 48 | Remate: 72 | Capacidade de Passe: 58 | Recuperacao: 99
39 | Joao Antunes Lourenco | Medio
Overall: 65 | Velocidade: 73 | Resistencia: 72 | Destreza: 51 | Impulsao: 71 | Jogo de Cabeça: 51 | Remate: 62 | Capacidade de Passe: 59 | Recuperacao: 75
2 | Francisca Teixeira Baptista | Defesa
Overall: 50 | Velocidade: 27 | Resistencia: 63 | Destreza: 93 | Impulsao: 48 | Jogo de Cabeça: 50 | Remate: 3 | Capacidade de Passe: 51 | Marcação: 50
1 | Maria Sofia da Rocha Gomes | Avançado
Overall: 62 | Velocidade: 55 | Resistencia: 68 | Destreza: 56 | Impulsao: 92 | Jogo de Cabeça: 27 | Remate: 69 | Capacidade de Passe: 89 | Posicionamento: 50
32 | Joao Duarte Vila Verde Silva | Avançado
Overall: 61 | Velocidade: 94 | Resistencia: 62 | Destreza: 58 | Impulsao: 84 | Jogo de Cabeça: 72 | Remate: 16 | Capacidade de Passe: 69 | Posicionamento: 50

```

### 5.2.3 Banco

```

*** Banco ***
0 | Rui Filipe Coelho Moreira | Medio
Overall: 59 | Velocidade: 92 | Resistencia: 31 | Destreza: 88 | Impulsao: 75 | Jogo de Cabeça: 65 | Remate: 66 | Capacidade de Passe: 67 | Recuperacao: 36
3 | Gonçalo da Ponte Carvalho | Medio
Overall: 57 | Velocidade: 72 | Resistencia: 51 | Destreza: 71 | Impulsao: 61 | Jogo de Cabeça: 68 | Remate: 74 | Capacidade de Passe: 0 | Recuperacao: 90
35 | Teresa Costa Pires Gil Fortes | Defesa
Overall: 57 | Velocidade: 85 | Resistencia: 29 | Destreza: 39 | Impulsao: 62 | Jogo de Cabeça: 56 | Remate: 83 | Capacidade de Passe: 69 | Marcação: 50
4 | Jose Diogo Martins Vieira | Guarda-Redes
Overall: 60 | Velocidade: 85 | Resistencia: 50 | Destreza: 51 | Impulsao: 62 | Jogo de Cabeça: 66 | Remate: 6 | Capacidade de Passe: 57 | Elasticidade: 69
37 | Ivo Miguel Gomes Lima | Avançado
Overall: 60 | Velocidade: 84 | Resistencia: 69 | Destreza: 75 | Impulsao: 57 | Jogo de Cabeça: 30 | Remate: 72 | Capacidade de Passe: 45 | Posicionamento: 50
14 | Matilde Oliveira Pizarro Bravo | Medio
Overall: 58 | Velocidade: 51 | Resistencia: 73 | Destreza: 69 | Impulsao: 49 | Jogo de Cabeça: 81 | Remate: 72 | Capacidade de Passe: 20 | Recuperacao: 67
50 | Gonçalo Pereira Figueiredo Ferreira | Avançado
Overall: 50 | Velocidade: 51 | Resistencia: 72 | Destreza: 58 | Impulsao: 38 | Jogo de Cabeça: 63 | Remate: 0 | Capacidade de Passe: 98 | Posicionamento: 50
21 | Marlon Daniel Duarte Ferreira | Lateral
Overall: 60 | Velocidade: 54 | Resistencia: 89 | Destreza: 68 | Impulsao: 87 | Jogo de Cabeça: 70 | Remate: 2 | Capacidade de Passe: 57 | Cruzamento: 59
27 | Rui Pedro Gomes Coelho | Defesa
Overall: 50 | Velocidade: 68 | Resistencia: 31 | Destreza: 68 | Impulsao: 57 | Jogo de Cabeça: 27 | Remate: 70 | Capacidade de Passe: 56 | Marcação: 50

```

### 5.2.4 Substituição

Deseja Fazer Alterações?

1. Sim

0. Não

Opção: 1

Selecione Jogador dos Titulares

Digite Numero: 39

Selecione Jogador dos Titulares/Banco

Digite Numero: 35

```

*** Banco ***
0 | Rui Filipe Coelho Moreira | Medio
Overall: 59 | Velocidade: 92 | Resistencia: 31 | Destreza: 88 | Impulsao: 75 | Jogo de Cabeça: 65 | Remate: 66 | Capacidade de Passe: 67 | Recuperacao: 36
3 | Gonçalo da Ponte Carvalho | Medio
Overall: 57 | Velocidade: 72 | Resistencia: 51 | Destreza: 71 | Impulsao: 61 | Jogo de Cabeça: 68 | Remate: 74 | Capacidade de Passe: 0 | Recuperacao: 90
4 | Jose Diogo Martins Vieira | Guarda-Redes
Overall: 60 | Velocidade: 85 | Resistencia: 50 | Destreza: 51 | Impulsao: 62 | Jogo de Cabeça: 66 | Remate: 6 | Capacidade de Passe: 57 | Elasticidade: 69
37 | Ivo Miguel Gomes Lima | Avançado
Overall: 60 | Velocidade: 84 | Resistencia: 69 | Destreza: 75 | Impulsao: 57 | Jogo de Cabeça: 30 | Remate: 72 | Capacidade de Passe: 45 | Posicionamento: 50
39 | Joao Antunes Lourenco | Medio
Overall: 65 | Velocidade: 73 | Resistencia: 72 | Destreza: 51 | Impulsao: 71 | Jogo de Cabeça: 51 | Remate: 62 | Capacidade de Passe: 59 | Recuperacao: 75
14 | Matilde Oliveira Pizarro Bravo | Medio
Overall: 58 | Velocidade: 51 | Resistencia: 73 | Destreza: 69 | Impulsao: 49 | Jogo de Cabeça: 81 | Remate: 72 | Capacidade de Passe: 20 | Recuperacao: 67
50 | Gonçalo Pereira Figueiredo Ferreira | Avançado
Overall: 50 | Velocidade: 51 | Resistencia: 72 | Destreza: 58 | Impulsao: 38 | Jogo de Cabeça: 63 | Remate: 0 | Capacidade de Passe: 98 | Posicionamento: 50
21 | Marlon Daniel Duarte Ferreira | Lateral
Overall: 60 | Velocidade: 54 | Resistencia: 89 | Destreza: 68 | Impulsao: 87 | Jogo de Cabeça: 70 | Remate: 2 | Capacidade de Passe: 57 | Cruzamento: 59
27 | Rui Pedro Gomes Coelho | Defesa
Overall: 50 | Velocidade: 68 | Resistencia: 31 | Destreza: 68 | Impulsao: 57 | Jogo de Cabeça: 27 | Remate: 70 | Capacidade de Passe: 56 | Marcação: 50

```



```

*** Titulares ***
30 | Rafael Diogo Batista Carvalho | Guarda-Redes
Overall: 64 | Velocidade: 67 | Resistencia: 52 | Destreza: 51 | Impulsao: 81 | Jogo de Cabeça: 14 | Remate: 67 | Capacidade de Passe: 68 | Elasticidade: 70
34 | Afonso Brandao Cruz | Defesa
Overall: 60 | Velocidade: 42 | Resistencia: 82 | Destreza: 50 | Impulsao: 58 | Jogo de Cabeça: 71 | Remate: 57 | Capacidade de Passe: 75 | Marcação: 50
24 | Lucas da Silva Carvalho | Defesa
Overall: 57 | Velocidade: 53 | Resistencia: 72 | Destreza: 19 | Impulsao: 52 | Jogo de Cabeça: 96 | Remate: 21 | Capacidade de Passe: 56 | Marcação: 50
36 | Diogo Francisco Lima Barros | Lateral
Overall: 71 | Velocidade: 70 | Resistencia: 70 | Destreza: 81 | Impulsao: 33 | Jogo de Cabeça: 9 | Remate: 83 | Capacidade de Passe: 72 | Cruzamento: 73
47 | Gonçalo Joao de Oliveira Barbosa Lobarinhas Henriques | Lateral
Overall: 66 | Velocidade: 98 | Resistencia: 92 | Destreza: 54 | Impulsao: 90 | Jogo de Cabeça: 56 | Remate: 75 | Capacidade de Passe: 73 | Cruzamento: 14
5 | Rodrigo Caldas Meira | Medio
Overall: 74 | Velocidade: 96 | Resistencia: 53 | Destreza: 91 | Impulsao: 84 | Jogo de Cabeça: 63 | Remate: 78 | Capacidade de Passe: 94 | Recuperacao: 54
43 | Andre Pizarro Martins | Medio
Overall: 66 | Velocidade: 63 | Resistencia: 57 | Destreza: 56 | Impulsao: 53 | Jogo de Cabeça: 48 | Remate: 72 | Capacidade de Passe: 58 | Recuperacao: 99
35 | Teresa Costa Pires Gil Fortes | Defesa
Overall: 57 | Velocidade: 85 | Resistencia: 29 | Destreza: 39 | Impulsao: 62 | Jogo de Cabeça: 56 | Remate: 83 | Capacidade de Passe: 69 | Marcação: 50
2 | Francisca Teixeira Baptista | Defesa
Overall: 50 | Velocidade: 27 | Resistencia: 63 | Destreza: 93 | Impulsao: 48 | Jogo de Cabeça: 50 | Remate: 3 | Capacidade de Passe: 51 | Marcação: 50
1 | Maria Sofia da Rocha Gomes | Avançado
Overall: 62 | Velocidade: 55 | Resistencia: 68 | Destreza: 56 | Impulsao: 92 | Jogo de Cabeça: 27 | Remate: 69 | Capacidade de Passe: 89 | Posicionamento: 50
32 | Joao Duarte Vila Verde Silva | Avançado
Overall: 61 | Velocidade: 94 | Resistencia: 62 | Destreza: 58 | Impulsao: 84 | Jogo de Cabeça: 72 | Remate: 16 | Capacidade de Passe: 69 | Posicionamento: 50

```

## 5.2.5 Substituição Programada

```

*** Substituições Programadas ***

```

```

Deseja Programar Alterações?

```

```

1. Sim

```

```

0. Não

```

```

Opção: 1

```

```

Selecione Primeiro Jogador (Titulares)

```

```

Digite Numero: 32

```

```

Selecione Segundo Jogador

```

```

Digite Numero: 14

```

```

Escolha Momento

```

```

1. Primeira Parte

```

```

2. Segunda Parte

```

```

0. Intervalo

```

```

Opção: 2

```

```

*** Substituições Programadas ***

```

```

Segunda Parte: 32 -> 14

```

```

Deseja Programar Alterações?

```

```

1. Sim

```

```

0. Não

```

```

Opção:

```

### 5.2.5 Escolha Tática

```
*** Modelo tatico ***  
Overall Equipa: 62  
1. 442 *  
2. 433  
3. 352  
0. Prosseguir  
Opção: 3  
  
*** Modelo tatico ***  
Overall Equipa: 57  
1. 442  
2. 433  
3. 352 *  
0. Prosseguir  
Opção:
```

### 5.2.6 Simulação do Jogo

```
Início da Partida  
  
Ataque: Bach F. C.  
  
Pressione Enter Para Continuar!  
  
Perda de Bola  
  
Bach F. C. 0 : 0 Bartok F. C.  
  
Pressione Enter Para Continuar!  
  
Ataque: Bartok F. C.  
  
Pressione Enter Para Continuar!  
  
Golo!!!  
  
Bach F. C. 0 : 1 Bartok F. C.
```

### 5.2.7 Cálculo do Resultado

```
Resultado Final  
Bach F. C. 0 : 2 Beethoven F. C.
```

### 5.3 Menu Jogos Realizados

```
*** Selecione Jogo ***  
1. Sporting Club Shostakovich - Mendelssohn F. C.  
2. Mozart F. C. - Sporting Club Dvorak  
3. Debussy Athletic - Stravinsky Athletic  
4. Schumann Athletic - Beethoven F. C.  
5. Mendelssohn F. C. - Bartok F. C.  
6. Bach F. C. - Sporting Club Shostakovich
```

```
Data: 2021-03-30 | Terminado | Sporting Club Shostakovich 0 : 0 Mendelssohn F. C.
```

```
*** Alinhamentos Finais ***
```

```
Sporting Club Shostakovich
```

```
[25, 30, 1, 37, 33, 11, 38, 31, 39, 6, 12]
```

```
Mendelssohn F. C.
```

```
[2, 42, 40, 16, 25, 31, 41, 17, 45, 33, 36]
```

```
*** Substituições ***
```

```
Sporting Club Shostakovich
```

```
25 -> 3    43 -> 25    22 -> 37
```

```
Mendelssohn F. C.
```

```
1 -> 42    49 -> 31    14 -> 45
```

## 5.4 Menu Jogadores

```
*** Seleccione Opção ***  
1. Consultar Jogador  
2. Transferir Jogador  
3. Criar Jogador  
4. Apagar Jogador  
0. Sair  
Opção: 1
```

### 5.4.1 Consultar Jogador

```
Jose Pedro Gomes Ferreira | Posição: Médio  
Clube Atual: Debussy Athletic | Clubes Anteriores:  
Habilidade: 55 | Velocidade: 69 | Resistência: 58 | Destreza: 91 | Impulsão: 63 | Jogo de Cabeça: 52 | Remate: 41 | Capacidade de Passe: 50 | Recuperação: 33  
Pressione Enter Para Continuar!
```

```
Rui Jordao Sampaio Goncalves | Posição: Guarda-Redes  
Clube Atual: Schumann Athletic | Clubes Anteriores:  
Habilidade: 59 | Velocidade: 61 | Resistência: 86 | Destreza: 53 | Impulsão: 60 | Jogo de Cabeça: 66 | Remate: 64 | Capacidade de Passe: 61 | Elasticidade: 52  
Pressione Enter Para Continuar! |
```

```
Tiago Andre Oliveira Leite | Posição: Lateral  
Clube Atual: Wagner Athletic | Clubes Anteriores:  
Habilidade: 51 | Velocidade: 56 | Resistência: 68 | Destreza: 76 | Impulsão: 78 | Jogo de Cabeça: 79 | Remate: 65 | Capacidade de Passe: 31 | Cruzamento: 1  
Pressione Enter Para Continuar!
```

### 5.4.2 Transferir Jogador

```
358. Vasco Oliveira Matos  
359. Vicente Goncalves Moreira  
360. Vítor Lelis Noronha Leite  
0. Sair  
Opção: 359
```

```
16. Stravinsky Athletic  
17. Vivaldi F. C.  
18. Wagner Athletic  
0. Sair  
Opção: 17
```

```
Vicente Goncalves Moreira | Posição: Guarda-Redes  
Clube Atual: Vivaldi F. C. | Clubes Anteriores: Sporting Club Dvorak  
Habilidade: 43 | Velocidade: 2 | Resistência: 63 | Destreza: 66 | Impulsão: 54 | Jogo de Cabeça: 59 | Remate: 81 | Capacidade de Passe: 51 | Elasticidade: 16  
Pressione Enter Para Continuar! |
```

### 5.4.3 Criar Jogador

```
Digite Nome: Ronaldo
1: Guarda-Redes:
2: Defesa:
3: Lateral:
4: Medio:
5: Avancado:
Escolha Posicao: 5
Digite Numero: 7
Digite Valor De Velocidade: 80
Digite Valor De Resistencia: 60
Digite Valor De Agilidade: 90
Digite Valor De Impulsao: 90
Digite Valor De Jogo de Cabeça: 100
Digite Valor De Capacidade de Remate: 90
Digite Valor De Capacidade de Passe: 70
Digite Valor De Posicionamento: 90
```

```
Ronaldo | Posição: Avançado
Clube Atual: Sem Equipa | Clubes Anteriores:
Habilidade: 85 | Velocidade: 80 | Resistência: 60 | Destreza: 90 | Impulsão: 90 | Jogo de Cabeça: 100 | Remate: 90 | Capacidade de Passe: 70 | Posicionamento: 90
Pressione Enter Para Continuar!
```

### 5.4.4 Apagar Jogador

```
308. Rodrigo Pires Rodrigues
309. Rodrigo Simoes Guimaraes Mendes
310. Ronaldo
311. Ruben Cesar Ferreira Lucas
312. Ruben Daniel Almeida Adao
```

```
308. Rodrigo Pires Rodrigues
309. Rodrigo Simoes Guimaraes Mendes
310. Ruben Cesar Ferreira Lucas
311. Ruben Daniel Almeida Adao
312. Ruben Filipe Martins Machado
313. Ruben Samuel Alves Santos
```

## 5.5 Menu Equipas

```
*** Seleccione Opção ***  
1. Consultar Equipa  
2. Criar Equipa  
3. Apagar Equipa  
0. Sair  
Opção:
```

### 5.6.1 Consultar Equipa

```
Bach F. C.  
Overall: 60  
  
Plantel:  
  
0 | Rui Filipe Coelho Moreira | Medio  
Overall: 59 | Velocidade: 92 | Resistencia: 31 | Destreza: 88 | Impulsao: 75 | Jogo de Cabeça: 65 | Remate: 66 | Capacidade de Passe: 67 | Recuperacao: 36  
  
1 | Maria Sofia da Rocha Gomes | Avançado  
Overall: 62 | Velocidade: 55 | Resistencia: 68 | Destreza: 56 | Impulsao: 92 | Jogo de Cabeça: 27 | Remate: 69 | Capacidade de Passe: 89 | Posicionamento: 50  
  
2 | Francisca Teixeira Baptista | Defesa  
Overall: 50 | Velocidade: 27 | Resistencia: 63 | Destreza: 93 | Impulsao: 48 | Jogo de Cabeça: 50 | Remate: 3 | Capacidade de Passe: 51 | Marcação: 50  
  
3 | Goncalo da Ponte Carvalho | Medio  
Overall: 57 | Velocidade: 72 | Resistencia: 51 | Destreza: 71 | Impulsao: 61 | Jogo de Cabeça: 68 | Remate: 74 | Capacidade de Passe: 0 | Recuperacao: 90
```

### 5.6.2 Criar Equipa

```
Digite Nome: Braga
```

```
*** Seleccione Equipa ***  
1. Bach F. C.  
2. Bartok F. C.  
3. Beethoven F. C.  
4. Braga
```

### 5.6.3 Apagar Equipa

```
Opção: 4

*** Seleccione Equipa ***
1. Bach F. C.
2. Bartok F. C.
3. Beethoven F. C.
4. Brahms F. C.
5. Debussy Athletic
```

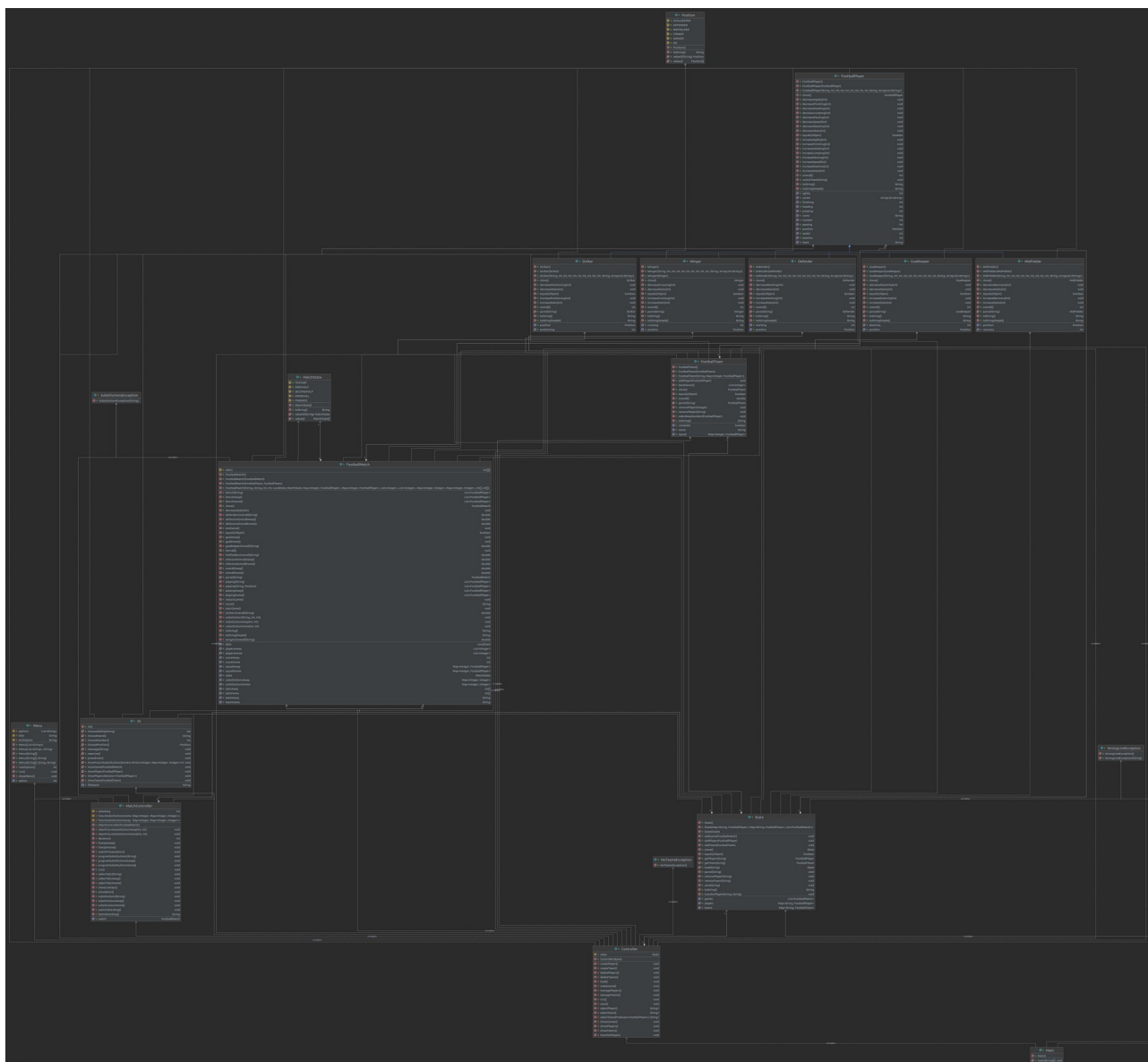
### 5.7 Guardar Estado

```
Digite Nome Do Ficheiro: teste
Guardado Com Sucesso!
```

### 5.8 Carregar Estado

```
Digite Nome Do Ficheiro: teste
Carregado Com Sucesso!
```

## 6. Diagrama de Classes



*Diagrama de classes do programa, gerado pelo IntelliJ.*



## **7. Conclusão**

Com o projeto concluído esperamos ter cumprido todos os requisitos que nos foram propostos e que o nosso programa respeite os princípios da programação orientada a objetos.

Um dos fatores que gostaríamos de ter melhorado era a implementação de uma arquitetura model-view-controller semelhante à que nos foi apresentada tanto nas aulas teóricas como nas práticas, em que o controller é uma das variáveis de instância da view. Tal não foi feito pois não sabíamos como fazer este tipo de implementação pelo que optamos por arquitetura na qual nos sentíamos mais confortáveis.

Foi o nosso primeiro projeto de média escala realizado em linguagem Java e todos concordamos que o mesmo foi importante para consolidar os conhecimentos adquiridos ao longo do semestre.