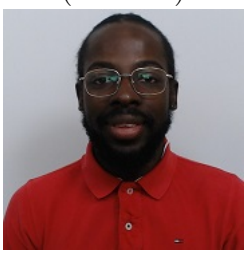
Programação Orientada aos Objetos

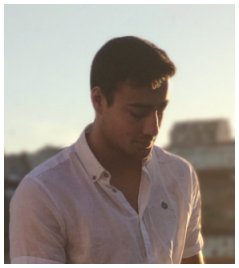
Trabalho Prático - Football Manager

Relatório de Desenvolvimento - Grupo 52

Áureo Joel Costa dos Santos Benedito
(A76068)



João Aniceto Rodrigues Rocha
(A80292)



João Guilherme Rodrigues Reis
(A84671)

June 12, 2021

1 Introduction

No âmbito da unidade curricular de Programação Orientada aos Objetos foi-nos proposto a realização de um tabalho prático de forma a podermos aplicar o conhecimento adquirido nas aulas. Neste trabalho iremos aplicar os conhecimentos para resolvermos a tarefa que nos foi dada.
O projeto solicitado passa por construir uma aplicação em Java, em criar uma aplicação que permita simular uma partida de futebol.

2 Documentação das classes

2.1 Jogador

A classe Jogador é uma super classe abstrata responsável pelo polimorfismo do programa. Se por exemplo uma subclasse Avançado for instanciada, terá toda a herança da classe Jogador, incluindo variáveis de instância, tal como os seus métodos. Utilizamos esta dinâmica para diferenciar os jogadores entre si pela posição que jogam em campo. O principal identificador de cada jogador é o seu número de camisola, um dos desafios da implementação desta estratégia é a preocupação com dois jogadores terem o mesmo nome, mas no mundo real do futebol, não é permitido na mesma equipa ter dois jogadores com o mesmo número. Logo para efeito de endereçamento de um jogador fazemos a seguinte chamada:

```
Defesa d = new Defesa();
Equipa Porto = new Equipa();
Porto.addJogador(d);
System.out.println("Objeto jogador: "+Porto.getEquipa().get(d.getNum()));
```

As variáveis de instância da classe Jogador são as seguintes.

```
public abstract class Jogador implements Comparable<Jogador>, Serializable {
    private int numero;
    private String nome;
    private boolean titular;
    private ArrayList<Equipa> historial;
    private int velocidade;
    private int resistencia;
    private int destreza;
    private int impulsao;
    private int jogodecabeca;
    private int remate;
    private int passe;
    private double habilidade;
```

No que toca à titularidade de um jogador, à semelhança da vida real, apenas é definido se um jogador vai jogar a titular ou não nos momentos antes de começar um jogo, ou seja apenas quando o jogo é iniciado os valores de titular de cada jogador da equipa são alterados. O historial de um jogador é alterado dinamicamente se houver uma transferência de jogador entre equipas, e o programa suporta também equipas que não estejam definidas a priori. Este é o método que se encarrega de atualizar o historial de um jogador:

```
    public void atualizaHist(Equipa eq) {
        this.historial.add(eq.clone());
    }
```

2.1.1 GuardaRedes

A classe GuardaRedes é uma subclasse de Jogador, tem tudo que um "Jogador" tem, excepto a elasticidade.Foi criada da seguinte maneira:

```
public class GuardaRedes extends Jogador {
    private int elasticidade;
    }
```

2.1.2 Defesa

Nesta classe foi criada, além de tudo o que um "Jogador" tem, a variavel "corte".

```
public class Defesa extends Jogador {
    private int corte;
    }
```

2.1.3 Médio

Na classe Medio criamos a variável "recuperação".

```
public class Medio extends Jogador {
    private int recup;
    }
```

2.1.4 Lateral

Na classe foi criada a variável "cruzamento".

```
public class Lateral extends Jogador {
    private int cruzamento;
    }
```

2.1.5 Avançado

À classe Avancado foi criada a variavel drible.

```
public class Avançado extends Jogador {
    private int drible;
    }
```

2.2 Equipa

Para formar uma equipa de jogadores utilizamos apenas duas variáveis de instância. A primeira é uma string com o nome da Equipa, a segunda uma *HashMap* com o número de camisola dos jogadores como *key*, e o objeto jogador correspondente como *value*.
O método que foi desenvolvido para adicionar jogadores consiste em adicionar o jogador ao map de instância e atualizar o historial do mesmo jogador.

```
    public void addJogador(Jogador o) {
        this.equipa.put(o.getNum(), o.clone());
        o.atualizaHist(this);
    }
```

Para remover um jogador da equipa (método muito útil na transferência de jogadores entre equipas), percorremos os *values* do *map*, caso o número do jogador da lista corresponder com o número do jogador de argumento, este é removido diretamente.

```
    public void removeJogador(Jogador o) {
        for (Jogador j : this.equipa.values()) {
            if (o.getNum() == j.getNum()) {
                this.equipa.remove(o.getNum());
                break;
            }
        }
    }
```

Listing 1: removeJogador

Seguindo a mesma lógica do método anterior, podemos construir a transferenciaJogador, como chamamos novamente o método addJogador, não nos precisamos de preocupar com atualizar o historial do jogador de entrada.

```
    //equipaAtual.transferenciaJogador(equipaAserTransferido,jogador da equipa atual)
    public void transferenciaJogador(Equipa a, Jogador o) {
        for (Jogador j : this.equipa.values()) {
            if (o.getNum() == j.getNum()) {
                removeJogador(o);
                a.addJogador(o);
                break;
            }
        }
    }
```

O método substituição troca apenas os valores de titularidade de dois jogadores, retorna mensagens de aviso caso algo corra mal, nomeadamente um dos jogadores não constar na equipa, ou estarem ambos em campo.

```
    //jogador t e b trocam os seus valores de Titular
    public void substituicao(Jogador t, Jogador b) {

        int flag = 0;
        boolean b1 = t.getTitular();
        boolean b2 = b.getTitular();
        for (Jogador j : this.equipa.values()) {
            if ((t.getNum() == j.getNum()) || (b.getNum() == j.getNum()))
                flag++;
        }
        if ((flag == 2) && (t.getTitular()!=b.getTitular())){
            t.setTitular(b2);
            b.setTitular(b1);
        }
        else {
            System.out.println("\nErro na substituição de jogadores (verifique se têm
                valor titular distinto ou se pertencem à mesma equipa)");
        }
        System.out.println(t.getNomeReduzido() + " vai ser substituido por: " + b.
            getNomeReduzido() + "\n");
    }
```

Posteriormente será necessário simular um jogo, onde o valor de habilidade de uma equipa influencia as jogadas efetuadas na simulação do jogo.

```
    public double gethabilidades() {
        double sum = 0;
        int i=0;
        for (Jogador j : this.equipa.values()) {
            if (j.getTitular()==true) {
                sum += j.gethabilidade("");

                i += 1;
            }
        }
        double media=0;
        media = sum/i;
        return media;
    }
```

2.3 Delegate

A classe Delegate é a *view* do nosso programa. Encarrega-se de criar vários menus e métodos que vão interagir diretamente com o utilizador. Recorre com frequência à classe Data que é por si o *controller* da aplicação, e tem uma variável de instância onde está guardada todas as estruturas de dados do simulador. Todos os menus utilizados são da classe Menu fornecida pelos docentes.

```
public class Delegate {
    //INSTANCE VARIABLES
    private Data info;
    private Menu menuPrincipal, menuSecundario, menuEquipas, menuJogos, menuCriacao;
    private Scanner scannerIn;
```

O output é o menu interativo seguinte.

```
 *** Menu ***
1 - Jogadores
2 - Equipas
3 - Jogos
4 - Efetuar Transferência
5 - Simular um Jogo
6 - Modo Criação
7 - Salvar Dados
8 - Carregar Dados
9 - Eliminar Dados
0 - Sair
Opção:
```

A primeira opção possibilita expor os nomes de todos os jogadores. Para efeitos logisticos esta lista de output foi encurtada.

```
(...)
Andre Goncalves Vieira
Vasco Mota de Oliveira
Joao Pedro Carvalho Henriques
Rafael dos Anjos Areas
Nuno Alexandre Pereira Machado
Joao Manuel Cardoso Guedes

 *** Menu ***
1 - Mais Informação
0 - Sair
Opção:
```

É seguidamente criado um sub-menu onde o utilizador pode escolher se pretende verificar a informação de todos os jogadores, incluindo Equipa, histórico, posicionamento, número, nível de habilidade, titularidade e valores de habilidade individuais.

```
(...)
Joao Manuel Cardoso Guedes (Avançado , Mendelssohn F. C.)

Numero do Jogador: 13
Habilidade: 48.050000000000004
É titular? false
Historial: Mendelssohn F. C.
Valor de Velocidade: 5
Valor de Destreza: 55
Valor de Impulsão: 52
Valor de Jogo de Cabeça: 67
Valor de Remate: 68
Valor de Capacidade de Passe: 1
```

O mesmo se aplica para as equipas.

```
LISTA DAS EQUIPAS

Schumann Athletic
Stravinsky Athletic
Bach F. C.
Debussy Athletic
Mozart F. C.
Handel Athletic
Mendelssohn F. C.
Sporting Club Shostakovich
Sporting Club Schubert
Sporting Club Chopin
Mahler Athletic
Bartok F. C.
Beethoven F. C.
Sporting Club Prokofiev
Vivaldi F. C.
```

```
Sporting Club Dvorak
Brahms F. C.
Wagner Athletic

 *** Menu ***
1 - Mais Informação
0 - Sair
Opção:
```

Se requisitarmos mais informação relativa a equipas, somos apresentados com uma lista (encurtada novamente devido a logistica) de jogadores de cada uma das equipas.

```
(...)
Brahms F. C.
Pedro Henrique PessÃ a  Camargo , Delio Miguel Lopes Alves , Andre Manuel da Costa
    Ferreira , Daniel Barbosa Faria , Andre da Silva Veiga , Joao Manuel Silva Antunes
    , Feliciano Jose Sousa Oliveira , Jose Pedro Pinheiro da Silva , Joel Magalhaes
    Pinto , Pedro Calheno Pinto , Francisco Maria Sousa Goncalves Paiva , Francisco
    Torrinha Martins Torres Machado , Joana Maia Teixeira Alves , Carlos Humberto da
    Silva Ferreira , Rodrigo Jorge Pinto Guimaraes , Rui Miguel Borges Braga , Vasco
    Mota Marques , Goncalo Duarte Gomes Rodrigues , Guilherme Santiago Lopes Pereira ,
    Alexandre Rodrigues Balde ,

Wagner Athletic
Rui Pedro Chaves Silva Lousada Alves , Miguel Angelo Ferreira Fernandes , Cristiano
    Jose Goncalves Neiva Pereira , Daniel Torres Neves , Pedro Santos Arezes Costa ,
    Luis Miguel Teixeira Fernandes , Guilherme de Araujo Soares , Andre Oliveira de
    Paula Boechat Morandi , Jose Joaquim Machado Barbosa , Alexandre Eduardo Vieira
    Martins , Pedro Alexandre Coutinho Goncalves de Sousa , Joao Manuel Silva de Amorim
     , David Alexandre Ferreira Duarte , Ricardo Manuel Soares Peixoto , Sara Lima
    Pereira , Tiago Andre Oliveira Leite , Beatriz Ribeiro Terra Almeida , Pedro
    Gabriel Fernandes Pereira , Alexandre Silva Martins , Joao Pedro Fontes Delgado ,
```

No que toca à terceira opção, os jogos, podemos também requisitar que seja apresentado no ecrã a info de todos os jogos. É importante referir que as ambas as equipas da casa como a equipa visitante não apresentam ainda definidos os modelos táticos, mas estes vão ser formados mal o jogo seja iniciado.

```
Equipa da casa: Stravinsky Athletic
Equipa visitante: Sporting Club Prokofiev
Data: 2021-06-12
Substituições da Equipa da Casa: [23, 24, 10](out)[36, 1, 23](in)
Substituições da Equipa Visitante: [1, 35, 30](out)[38, 47, 3](in)
Estado do jogo: por_iniciar
Golos da Equipa da Casa: 2
Golos da Equipa Visitante: 2
Modelo Tático Equipa Casa: null
Modelo Tático Equipa Visitante: null
Lista de Jogadores Titulares da Casa: [36, 29, 13, 19, 42, 1, 50, 22, 8, 17, 0]
Lista de Jogadores Titulares Visitantes: [3, 38, 15, 9, 5, 43, 28, 17, 47, 11, 48]
```

Para efetuar uma transfêrencia de um jogador entre duas equipas, é apresentada ao utilizador uma interface para que seja escolhida a equipa que contém o jogador que está de saída.

```
 *** Menu ***
1 - Jogadores
2 - Equipas
3 - Jogos
4 - Efetuar Transferência
5 - Simular um Jogo
6 - Modo Criação
7 - Salvar Dados
8 - Carregar Dados
9 - Eliminar Dados
0 - Sair
Opção: 4
Indique a equipa com o jogador de saída.
0 - Schumann Athletic
```

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

Seguidamente, é requisitado o inteiro necessário para selecionar a equipa com o jogador de entrada. Queremos fazer transferencia do jogador nº 36 da equipa *Schumann Athletic*, com o nome *Andre Filipe da Silva Marques*. E queremos que esse jogador seja inserido na equipa Sport Lisboa e Benfica.

```
Indique a equipa com o jogador de entrada.
0 - Schumann Athletic
1 - Stravinsky Athletic
2 - Bach F. C.
3 - Debussy Athletic
4 - Mozart F. C.
5 - Handel Athletic
6 - Sport Lisboa e Benfica
7 - Mendelssohn F. C.
8 - Sporting Club Shostakovich
9 - Sporting Club Schubert
10 - Sporting Club Chopin
11 - Mahler Athletic
12 - Bartok F. C.
13 - Beethoven F. C.
14 - Sporting Club Prokofiev
15 - Vivaldi F. C.
16 - Sporting Club Dvorak
17 - Brahms F. C.
18 - Wagner Athletic
Opção: 6

Indique o número do Jogador
36
```

Se voltarmos a pedir ao programa a informação sobre as equipas, conseguimos verificar que este jogador foi inserido com sucesso na equipa.

```
(...)
Sport Lisboa e Benfica
Svilar , Jardel , Andr Almeida , Odysseas , Andre Filipe da Silva Marques , Everton ,
    Simao Sabrosa , Darwin , Waldschmidt , Tiago Araujo , Seferovic , Angel DiMaria ,
    Joao Ferreira , Chiquinho , Pizzi , Goncalo Ramos , Nicolas Otamendi
(...)
```

Tal como o seu historial foi atualizado.

```
(...)
Andre Filipe da Silva Marques (GuardaRedes , Sport Lisboa e Benfica)

Numero do Jogador: 36
Habilidade: 61.099999999999994
É titular? false
Historial: Schumann Athletic Sport Lisboa e Benfica
Valor de Velocidade: 87
Valor de Destreza: 13
Valor de Impulsão: 55
```

```
Valor de Jogo de Cabeça: 72
Valor de Remate: 65
Valor de Capacidade de Passe: 73
Valor de Elasticidade: 62
(...)
```

No que toca ao output da simulação de um jogo vamos escolher a opção nº 5 do menu, e escolher um jogo. Vamos escolher o jogo 66 - Stravisnky Athletic VS Sporting Club Prokofiev.

```
Opção: 5

BEM-VINDO À SIMULAÇÃO DE JOGOS
Por favor, escolha uma jogo:
0 - Sporting Club Shostakovich VS Mendelssohn F. C.
1 - Mozart F. C. VS Sporting Club Dvorak
2 - Debussy Athletic VS Stravinsky Athletic
3 - Schumann Athletic VS Beethoven F. C.
4 - Mendelssohn F. C. VS Bartok F. C.
5 - Bach F. C. VS Sporting Club Shostakovich
(...)
64 - Sporting Club Prokofiev VS Mahler Athletic
65 - Beethoven F. C. VS Brahms F. C.
66 - Stravinsky Athletic VS Sporting Club Prokofiev
66
Começa o jogo na casa do Stravinsky Athletic contra Sporting Club Prokofiev

Stravinsky Athletic tem habilidade de 66.28181818181817
Sporting Club Prokofiev tem hab de 66.28636363636363
(...)
Bola começa no meio campo nos pés de Jorge Frederico
Jorge Frederico perdeu a bola para Tiago Leitao
Tiago Leitao passa a bola para Leonardo Araujo
Leonardo Araujo perde a bola no meio campo para Joao Alexandre
Joao Alexandre Consegue avançar no terreno para a zona de perigo
Joao Alexandre Faz Cruzamentoooo para Bruno Filipe
Bruno Filipe consegue cabeceaaaaar
 Graaaaaaande defesa de Jose Duarte
Guarda redes faz passe longo para Tiago Leitao
Tiago Leitao passa para Claudia Peixoto
Remate Fortissimo de Claudia Peixoto
Mas grande defesa de Joao Andre
Guarda redes faz passe longo para Jorge Frederico
Jorge Frederico passa a bola para Joao Alexandre
Joao Alexandre Consegue avançar no terreno para a zona de perigo
Joao Alexandre perde a bola para Leonardo Araujo
Leonardo Araujo Consegue avançar no terreno para o meio campo
Leonardo Araujo faz um passe interior para Tiago Leitao
Tiago Leitao passa a bola para Leonardo Araujo
Leonardo Araujo perde a bola no meio campo para Joao Alexandre
Joao Alexandre Consegue avançar no terreno para a zona de perigo
Joao Alexandre Faz Cruzamentoooo para Bruno Filipe
Rui Alexandre consegue cortar a bola no ar
Bola foi roubada por Bruno Filipe
Bruno Filipe perdeu a bola para Rui Alexandre
Rui Alexandre passa a bola para Leonardo Araujo
Leonardo Araujo perdeu a bola Joao Alexandre na zona de perigo
Leonardo Araujo perde a bola para Leonardo Araujo
Leonardo Araujo Consegue avançar no terreno para o meio campo
Leonardo Araujo perde a bola no meio campo para Joao Alexandre
Joao Alexandre perde a bola no meio campo para Leonardo Araujo
Leonardo Araujo perde a bola no meio campo para Joao Alexandre
Joao Alexandre Consegue avançar no terreno para a zona de perigo
Joao Alexandre Faz Cruzamentoooo para Bruno Filipe
Rui Alexandre consegue cortar a bola no ar
Bola foi roubada por Bruno Filipe
Remate Fortissimo de Bruno Filipe
E é golooooooo do Bruno Filipe
(...)
```

O jogo decorre durante 45 segundos, ao intervalo são feitas as substituições.

```
(...)
Intervalo do jogo:

Jogo Chegou aos Intervalo
Resultado:
Stravinsky Athletic : 0
Sporting Club Prokofiev : 1
 Subsituições da equipa casa :

Jose Duarte vai ser substituido por : Joao Henrique

Tiago Leitao vai ser substituido por : Shahzod Yusupov

Joao Henrique vai ser substituido por : David Pereira

 Subsituições da equipa Visitante :
Henrique Jose vai ser substituido por : Paulo Miguel

Mariana Filipa vai ser substituido por : Hugo Henrique

Joao Andre vai ser substituido por : Joao Paulo

Bola começa no meio campo nos pés de Tiago Leitao
Tiago Leitao passa a bola para Leonardo Araujo
Leonardo Araujo perde a bola no meio campo para Joao Alexandre
Joao Alexandre perde a bola no meio campo para Leonardo Araujo
Leonardo Araujo perde a bola no meio campo para Joao Alexandre
(...)
```

No fim, são expostos os resultados no ecrã, uma simulação bem sucedida.

```
(...)
Leonardo Araujo perde a bola no meio campo para Joao Alexandre
Joao Alexandre perde a bola no meio campo para Leonardo Araujo
Jogo chegou ao fim
Resultado:
Stravinsky Athletic : 0
Sporting Club Prokofiev : 2
```

Vamos verificar como se adicionam entidades ao sistema. Começamos por adicionar um jogador.

```
MENU DE PERSONALIZAÇÃO
Por favor, escolha uma opção:

 *** Menu ***
1 - Criar Jogador
2 - Criar Equipa
3 - Criar Jogo
0 - Sair
Opção: 1
Qual é o número de camisola?

7
Qual é o nome do Jogador?

Cristiano Ronaldo
Joga como titular?

true
Qual é o seu historial?

Real Madrid F.C.
É GuardaRedes, Avançado, Lateral, Médio ou Defesa?

Avançado
Quais of valores de velocidade, resistência, destreza, impulsão, cabeceamento, remate,
     passe e drible por ordem?

100 100 100 100 100 100 100 100
```

```
Finalmente, qual é a sua equipa?

Juventus
Jogador criado com sucesso
```

Juventus será adicionado à *database*.

```
LISTA DAS EQUIPAS

Schumann Athletic
Stravinsky Athletic
Bach F. C.
Debussy Athletic
Mozart F. C.
Handel Athletic
Sport Lisboa e Benfica
Mendelssohn F. C.
Sporting Club Shostakovich
Sporting Club Schubert
Sporting Club Chopin
Juventus <--
Mahler Athletic
Bartok F. C.
Beethoven F. C.
Sporting Club Prokofiev
Vivaldi F. C.
Sporting Club Dvorak
Brahms F. C.
Wagner Athletic
```

E finalmente, a informação do CR7 será adicionada automáticamente também.

```
Cristiano Ronaldo (Avançado , Juventus)

Numero do Jogador: 7
Habilidade: 100.0
É titular? true
Historial: Real Madrid F.C. Juventus
Valor de Velocidade: 100
Valor de Destreza: 100
Valor de Impulsão: 100
Valor de Jogo de Cabeça: 100
Valor de Remate: 100
Valor de Capacidade de Passe: 100
```

Para as equipas temos:

```
MENU DE PERSONALIZAÇÃO
Por favor, escolha uma opção:

 *** Menu ***
1 - Criar Jogador
2 - Criar Equipa
3 - Criar Jogo
0 - Sair
Opção: 2
Qual o nome da equipa?

A.C. Milan
```

Foi acrescentada a equipa A.C. Milan sem jogadores.

```
LISTA DAS EQUIPAS

Schumann Athletic
Stravinsky Athletic
Bach F. C.
Debussy Athletic
```

```
Mozart F. C.
Handel Athletic
Sport Lisboa e Benfica
Mendelssohn F. C.
Sporting Club Shostakovich
Sporting Club Schubert
Sporting Club Chopin
Juventus
Mahler Athletic
Bartok F. C.
Beethoven F. C.
Sporting Club Prokofiev
Vivaldi F. C.
Sporting Club Dvorak
Brahms F. C.
A.C. Milan
Wagner Athletic
```

Vamos agora criar um jogo com uma das equipas introduzidas pelo utilizador.

```
MENU DE PERSONALIZAÇÃO
Por favor, escolha uma opção:

 *** Menu ***
1 - Criar Jogador
2 - Criar Equipa
3 - Criar Jogo
0 - Sair
Opção: 3
Qual é a equipa da casa?

Sport Lisboa e Benfica
Quais são as substituições da equipa da casa?
Jogadores que entram:
34 1 10
Jogadores que saem:
30 99 88
Qual é a equipa Visitante?

Sporting Club Shostakovich
Quais são as substituições da equipa Visitante?
Jogadores que entram:
22 43 25
Jogadores que saem:
37 25 3
Quais os 11 titulares da equipa da Casa?

71 80 7 99 30 82 33 21 14 9 88
Quais os 11 titulares da equipa Visitante?

43 30 1 22 33 11 38 31 39 6 12
```

Conferimos a infromação do jogo.

```
Equipa da casa: Sport Lisboa e Benfica
Equipa visitante: Sporting Club Shostakovich
Data: 2021-06-13
Substituições da Equipa da Casa: [30, 99, 88](out)[34, 1, 10](in)
Substituições da Equipa Visitante: [37, 25, 3](out)[22, 43, 25](in)
Estado do jogo: por_iniciar
Golos da Equipa da Casa: 0
Golos da Equipa Visitante: 0
Modelo Tático Equipa Casa: null
Modelo Tático Equipa Visitante: null
Lista de Jogadores Titulares da Casa: [71, 80, 7, 99, 30, 82, 33, 21, 14, 9, 88]
Lista de Jogadores Titulares Visitantes: [43, 30, 1, 22, 33, 11, 38, 31, 39, 6, 12]
```

Vamos simular então o jogo adicionado.

```
(...)
64 - Sporting Club Prokofiev VS Mahler Athletic
65 - Beethoven F. C. VS Brahms F. C.
66 - Stravinsky Athletic VS Sporting Club Prokofiev
67 - Sport Lisboa e Benfica VS Sporting Club Shostakovich
67
Começa o jogo na casa do Sport Lisboa e Benfica contra Sporting Club Shostakovich

Sport Lisboa e Benfica tem habilidade de 66.25
Sporting Club Shostakovich tem hab de 67.47272727272728
(...)
```

E o resultado é:

```
(...)
Angel DiMaria Consegue avançar no terreno para a zona de perigo
Angel DiMaria Faz Cruzamentoooo para Darwin
Rodrigo Simoes consegue cortar a bola no ar
Jogo chegou ao fim
Resultado:
Sport Lisboa e Benfica : 1
Sporting Club Shostakovich : 1
```

2.4 Jogo

A classe jogo tem um dos métodos mais importantes do nosso projeto. Este método é responsavel por coordenar todas as instruções necessárias para a boa simulação de um jogo. Utilizamos o método Timer() para criar um contador de segundos, que a cada segundo gera fases de jogo.

```
        public void startJogo() {
         setTitularesCasa(this.titularesCasa);
         setTitularesVisitante(this.titularesVisitante);
         this.toString();
         System.out.println("Começa o jogo na casa do " + this.equipaCasa.getNome()+ "
             contra " + this.equipaVisitante.getNome() + "\n" );
         System.out.println(this.gethab());
         Timer timer = new Timer();
         timer.schedule(new TimerTask() {
             double coin = Math.random();

             Fase fase = new Fase(equipaCasa,equipaVisitante,0,coin >0.5 ? 1 : 0 ,4,
                 coin>0.5 ? equipaCasa.getJogador(modeloCasa.get("Medio").get(0)) :
                 equipaVisitante.getJogador(modeloVisitante.get("Medio").get(0)),
                 modeloCasa,modeloVisitante,entraCasa,saiCasa,entraVisitante,
                 saiVisitante );


             @Override
             public void run() {
                 fase.getState(fase.getEquipaAtacante(),fase.getFase(),fase.getJogador
                     ());
                 fase.setTime(fase.getTime() + 1);
                 if (fase.getTime() == 90) {
                     timer.cancel();  //stop the timer
                     fase.result(90);
                 };
             }
         }, 0, 1000);//wait 0 ms before doing the action and do it every 1000ms (1
             second)

    }
```

2.5 Fase

A classe fase foi criada para separar a simulação do jogo da classe jogo, no âmbito de reduzir complexidade. Esta classe está encarregada de simular um jogo, onde o tempo em segundos corresponde a minutos num jogo real. A cada segundo acontece uma situação de jogo real, dependendo do jogador que tem a bola nos pés e de em que parte do campo o jogador está. Aos 45 segundos/min é o intervalo, e são realizadas as subsituições,aos 90 segundos o jogo termina e é dado o resultado. A classe recebe todos as variáveis de instáncia necessárias para que seja realizado a simulação de um jogo.

```
public class Fase {
    private int golosVisitante;
    private int golosCasa;
    private int equipaAtacante;
    private int comecou;
    private int fase;
    private int time;
    private Equipa equipaCasa;
    private Equipa equipaVisitante;
    private Jogador bola;
    private HashMap<String, List<Integer>> modeloCasa;
    private HashMap<String,List<Integer>> modeloVisitante;
    private ArrayList<Integer> entracasa;
    private ArrayList<Integer> saicasa;
    private ArrayList<Integer> entraVisitante;
    private ArrayList<Integer> saiVisitante;
```

Dentro desta classe é realizada as situações de jogo em que está definida uma função getfase() que recebe a equipa que está a atacar que é representado por um int 1 corresponde a equipa da casa e 0 corresponde a equipa atacante recebe também um int fase 1 correponde a zona do campo da parte defensiva, fase 2 corresponde a zona do meio campo e zona 3 corresponde a zona de perigo temos também uma fase 4 que é para começar o jogo no meio campo e fase 5 que é quando o avançado recebe um cruzamento do lateral.A bola começa com a bola nos pés de um médio na fase 4.

```
    public void getState(int equipaAtacante,int fase,Jogador bola){
            if (this.time == 45){
                intervalo();
                // troca de equipa a bola
                this.equipaAtacante = 1-this.comecou;

                this.fase = 4;
                Equipa com = this.equipaAtacante == 1 ? equipaCasa : equipaVisitante;
                HashMap<String, List<Integer>> bol = this.equipaAtacante == 1 ?
                    modeloCasa : modeloVisitante;
                Medio segpart = new Medio((Medio) com.getEquipa().get(bol.get("Medio").
                    get(0)));
                this.bola = segpart;

            }
            else{

        switch(bola.getClass().getName()) {
```

Apartir daqui é usado um switch para ver a posição do jogador que tem a bola e ver que opções cada jogador tem

2.5.1 Defesa

Caso do Defesa, é calculada um fator sorte + habilidade do jogador em que dependendo desse valor o defesa pode passar para o médio ou pode passar para o lateral que pode estar ao lado dele ou ainda pode perder a bola para o avançado que está a frente dele e troca a equipa que está a atacar.

```
   case "Defesa":
                double luckDF = Math.random() * bola.gethabilidade("GuardaRedes");
                if (luckDF > 15) {
                    //Passar para médio
                    Equipa troca = this.equipaAtacante == 1 ? this.equipaCasa : this.
                        equipaVisitante;
```

```
                HashMap<String, List<Integer>> modelo = this.equipaAtacante == 1 ?
                    this.modeloCasa : this.modeloVisitante;
                if (Math.random() > 0.5) {
                    Medio recetor = new Medio((Medio) troca.getEquipa().get(modelo
                        .get("Medio").get(0)));
                    System.out.println(this.bola.getNomeReduzido() + " passa a
                        bola para " + recetor.getNomeReduzido());
                    this.bola = recetor;
                    this.fase++;
                } else { //passe para lateral mais
                    Lateral rec = new Lateral((Lateral) troca.getEquipa().get(
                        modelo.get("Lateral").get(0)));
                    System.out.println(this.bola.getNomeReduzido() + " passa a
                        bola para " + rec.getNomeReduzido());
                    this.fase = 1;
                    this.bola = rec;
                }
            } else {
                //perde a bola
                this.equipaAtacante = 1 - this.equipaAtacante;
                Equipa troca = this.equipaAtacante == 1 ? this.equipaCasa : this.
                    equipaVisitante;
                HashMap<String, List<Integer>> modelo = this.equipaAtacante == 1 ?
                    this.modeloCasa : this.modeloVisitante;
                this.fase = 4 - this.fase;
                Avançado intercetor = new Avançado((Avançado) troca.getEquipa().
                    get(modelo.get("Avançado").get(0)));
                this.bola = intercetor;
                System.out.println("Bola foi roubada por " + this.bola.
                    getNomeReduzido());


            }
```

2.5.2 Medio

Caso do Medio, tal com o defesa é calculada um fator sorte + habilidade do jogador de seguida pode passar para o avançado ou pode passar para o lateral que pode estar ao lado dele ou ainda pode perder a bola para o médio que está a frente dele e troca a equipa que está a atacar . Se a fase for igual a 4 significa que a bola começa no meio campo

```
    case "Medio":
                double luckME = Math.random() * bola.gethabilidade("Medio");
                if (this.fase == 4)
                    System.out.println("Bola começa no meio campo nos pés de " + this.
                        bola.getNomeReduzido());

                if (luckME > 20) {
                    // caso passe
                    Equipa nova = this.equipaAtacante == 1 ? equipaCasa :
                        equipaVisitante;
                    HashMap<String, List<Integer>> a = this.equipaAtacante == 1 ?
                        modeloCasa : modeloVisitante;
                    if (Math.random() > 0.5) { //passe para Avançado

                        this.fase += 1;
                        Avançado recetordebola = new Avançado((Avançado) nova.
                            getJogador(a.get("Avançado").get(0)));

                        System.out.println(bola.getNomeReduzido() + " passa para " +
                            recetordebola.getNomeReduzido());
                        this.bola = recetordebola;
                    } else { // passe para lateral
                        Lateral latrec = new Lateral((Lateral) nova.getJogador(a.get("
                            Lateral").get(0)));
                        System.out.println(this.bola.getNomeReduzido() + " passa a
                            bola para " + latrec.getNomeReduzido());
                        this.bola = latrec;
                        this.fase = 2;
```

```
            }
        } else {
            //caso perde a bola
            this.equipaAtacante = 1 - this.equipaAtacante;
            Equipa nova = this.equipaAtacante == 1 ? equipaCasa :
                equipaVisitante;
            HashMap<String, List<Integer>> a = this.equipaAtacante == 1 ?
                modeloCasa : modeloVisitante;
            Medio roubou = new Medio((Medio) nova.getJogador(a.get("Medio").
                get(0)));
            System.out.println(bola.getNomeReduzido() + " perdeu a bola para "
                + roubou.getNomeReduzido());
            this.bola = roubou;
            this.fase = 2;
        }
```

2.5.3 Guarda redes

Caso do GuardaRedes, tal com o defesa é calculada um fator sorte + habilidade do jogador de seguida pode passar para o defesa mias próximo ou pode ainda tentar chutar para um médio

```
        case "GuardaRedes":
        double luckGr = Math.random() * bola.gethabilidade("GuardaRedes");
        Equipa ataque = this.equipaAtacante == 1 ? equipaCasa :
            equipaVisitante;
        HashMap<String, List<Integer>> modeloataque = this.equipaAtacante == 1
             ? modeloCasa : modeloVisitante;
        if (luckGr > 15) {
            // passar para médio por jogo de cabeça a funcionar
            this.bola = new Medio((Medio) ataque.getJogador(modeloataque.get("
                Medio").get(0)));
            System.out.println("Guarda redes faz passe longo para " + this.
                bola.getNomeReduzido());
            this.fase = 2;
        } else {
            // passar para defesa

            this.bola = new Defesa((Defesa) ataque.getJogador(modeloataque.get
                ("Defesa").get(0)));
            this.fase = 1;
            System.out.println("Guarda redes faz passe curto para " + this.
                bola.getNomeReduzido());

        }
```

2.5.4 Avançado

Caso do Avançado, tal com o defesa é calculada um fator sorte + habilidade do jogador de seguida se o int fase == 5 ele recebe um cruzamento do lateral e consegue cabecear ,dependo da destreza do guardaredes * sorte ele vai ou não conseguir defender. Noutro caso o avançado pode perder a bola para o defesa que está a frente dele e troca a equipa que está a atacar, ou pode ainda rematar. Dentro do remate pode ou não acertar na baliza , se não acertar a bola é do guarda redes, se acertar o guarda redes tem ou não probabilidade de defender consoante um fator sorte * destreza do guardaredes.

```
case "Avançado":
        double luckAv = Math.random() * bola.gethabilidade("Avançado");
        this.equipaAtacante = 1 - this.equipaAtacante;
        Equipa nova = this.equipaAtacante == 1 ? equipaCasa : equipaVisitante;
        HashMap<String, List<Integer>> a = this.equipaAtacante == 1 ?
            modeloCasa : modeloVisitante;
        if (luckAv > 15) {
            // caso em que o avançado passa por o defesa
            GuardaRedes guardaredes = new GuardaRedes((GuardaRedes) nova.
                getJogador(a.get("GuardaRedes").get(0)));

            if (this.fase == 5) {
```

```
                System.out.println(this.bola.getNomeReduzido() + "␣consegue␣
                    cabeceaaaaar␣");
                if (this.bola.getCab() * Math.random() > guardaredes.getDes()
                    * Math.random()) {
                    System.out.println("␣Que␣Golaçooooooooo␣de␣" + this.bola.
                        getNomeReduzido());
                    Medio novo = new Medio((Medio) nova.getEquipa().get(a.get(
                        "Medio").get(0)));
                    this.bola = novo;
                    this.fase = 4;
                } else {
                    System.out.println("␣Graaaaaaande␣defesa␣de␣" +
                        guardaredes.getNomeReduzido());
                    this.bola = guardaredes;
                    this.fase = 1;
                }
            } else if (Math.random() * 100 > this.bola.getRem()) {
                // caso chute e falhe a bola é do guarda redes;
                System.out.println("Remate␣de␣" + bola.getNomeReduzido() + "␣
                    saiu␣completamente␣ao␣lado␣");
                System.out.println("Bola␣na␣mão␣do␣guarda␣redes␣");
                this.bola = guardaredes;
            } else {
                // caso chute e acerte na baliza vamos ver se o guarda redes
                    defende
                System.out.println("Remate␣Fortissimo␣de␣" + bola.
                    getNomeReduzido());
                if (Math.random() * 100 > guardaredes.getElast()) {
                    System.out.println("Mas␣grande␣defesa␣de␣" + guardaredes.
                        getNomeReduzido());
                    this.bola = guardaredes;
                    this.fase = 1;

                } else {
                    //caso guardaredes nao defenda
                    System.out.println("E␣é␣golooooooo␣do␣" + this.bola.
                        getNomeReduzido());
                    // bola volta para o meio campo para um médio (falta
                        implementar)
                    Medio meio = new Medio((Medio) nova.getJogador(a.get("
                        Medio").get(0)));
                    this.bola = meio;
                    this.fase = 4;
                    if ((1 - this.equipaAtacante) == 1) {
                        this.golosCasa += 1;
                    } else {
                        this.golosVisitante += 1;
                    }
                }

            }
```

2.5.5 Lateral

Caso do Lateral, tal como os outros jogadores existe um fator que determina o que vai acontecer, calculado por habilidade do avançado * sorte. O lateral joga dependendo da fase de jogo isto significa que na fase 1 o lateral está na zona defensiva , fase 2 esta no meio campo ,fase 3 esta na zona de perigo. consoante estas pode passar para um jogador que está na mesma fase onde ele está ou seja se estiver na fase 1 pode passar para um defesa. Se estiver na fase 3 em vez de passar para o avançado faz um cruzamento para o avançado e é disputado um jogo de cabeça com o defesa e o avançado para ver quem fica com a bola. Fora isto o lateral pode subir no terreno da fase 1 pode ir para a 2 e da 2 para a 3. O lateral também pode perder a bola para o lateral que está a frente dele e troca a equipa que está a atacar.

```
    case "Lateral":
            Equipa atlateral = this.equipaAtacante == 1 ? equipaCasa :
                equipaVisitante;
```

```
HashMap<String, List<Integer>> modellar = this.equipaAtacante == 1 ?
    modeloCasa : modeloVisitante;
double luckLT = Math.random() * bola.gethabilidade("GuardaRedes");
switch (this.fase) {
    case 1:
        //caso anda com a bola para a frente
        if (luckLT > 20) {
            System.out.println(this.bola.getNomeReduzido() + "
                Consegue avançar no terreno para o meio campo");
            this.fase += 1;
        }
        //passa para defesa
        else if (luckLT > 15) {
            Defesa recebe = new Defesa((Defesa) atlateral.getEquipa().
                get(modellar.get("Defesa").get(0)));
            System.out.println(this.bola.getNomeReduzido() + " faz
                passe interior para " + recebe.getNomeReduzido());
            this.bola = recebe;
        }
        //caso em que perde a bola
        else {
            this.equipaAtacante = 1 - this.equipaAtacante;
            atlateral = this.equipaAtacante == 1 ? equipaCasa :
                equipaVisitante;
            modellar = this.equipaAtacante == 1 ? modeloCasa :
                modeloVisitante;
            Lateral def = new Lateral((Lateral) atlateral.getEquipa().
                get(modellar.get("Lateral").get(0)));
            System.out.println(this.bola.getNomeReduzido() + " perdeu
                a bola " + def.getNomeReduzido() + " na zona de perigo
                ");
            this.fase = 3;

        }
        break;
    case 2:
        //caso anda com a bola para a frente
        if (luckLT > 20) {
            System.out.println(this.bola.getNomeReduzido() + "
                Consegue avançar no terreno para a zona de perigo");
            this.fase += 1;
        }
        //passa para médio
        else if (luckLT > 15) {
            Medio rece = new Medio((Medio) atlateral.getEquipa().get(
                modellar.get("Medio").get(0)));
            System.out.println(this.bola.getNomeReduzido() + " faz um
                passe interior para " + rece.getNomeReduzido());
            this.bola = rece;

        }
        //caso em que perde a bola
        else {
            this.equipaAtacante = 1 - this.equipaAtacante;
            atlateral = this.equipaAtacante == 1 ? equipaCasa :
                equipaVisitante;
            modellar = this.equipaAtacante == 1 ? modeloCasa :
                modeloVisitante;
            Lateral cort = new Lateral((Lateral) atlateral.getEquipa()
                .get(modellar.get("Lateral").get(0)));
            System.out.println(this.bola.getNomeReduzido() + " perde a
                bola no meio campo para " + cort.getNomeReduzido());
            this.bola = cort;


        }
        break;
    case 3:
        //caso faz cruzamento
        if (luckLT > 15) {
            Avançado cabeca = new Avançado((Avançado) atlateral.
```

```
                        getEquipa().get(modellar.get("Avançado").get(0)));
                    System.out.println(this.bola.getNomeReduzido() + " Faz
                        Cruzamentoooo para " + cabeca.getNomeReduzido());
                    atlateral = !(this.equipaAtacante == 1) ? equipaCasa :
                        equipaVisitante;
                    modellar = !(this.equipaAtacante == 1) ? modeloCasa :
                        modeloVisitante;
                    Defesa def = new Defesa((Defesa) atlateral.getEquipa().get
                        (modellar.get("Defesa").get(0)));
                    // caso avançado consiga cabecear
                    if (Math.random() * cabeca.getCab() > Math.random() * def.
                        getCab()) {
                        this.fase = 5;
                        this.bola = cabeca;

                        //caso defesa consiga defender
                    } else {
                        this.bola = def;
                        this.equipaAtacante = 1 - this.equipaAtacante;
                        System.out.println(this.bola.getNomeReduzido() + "
                            consegue cortar a bola no ar");
                    }

                    //caso em que perde a bola
                } else {
                    this.equipaAtacante = 1 - this.equipaAtacante;
                    atlateral = this.equipaAtacante == 1 ? equipaCasa :
                        equipaVisitante;
                    modellar = this.equipaAtacante == 1 ? modeloCasa :
                        modeloVisitante;

                    Lateral cut = new Lateral((Lateral) atlateral.getEquipa().
                        get(modellar.get("Lateral").get(0)));
                    System.out.println(this.bola.getNomeReduzido() + " perde a
                        bola para " + cut.getNomeReduzido());
                    this.bola = cut;
                    this.fase = 1;

                }

                break;
            }
```

3 Conclusão

O programa realizado é um simulador de de jogos de futebol, onde o utilizador se põe nos pés de um treinador de primeira liga com múltiplos jogadores à sua disposição, a fim de criar a melhor equipa. Este jogo é capaz de criar jogadores com diversas habilidades, e formar equipas com diferentes táticas de jogo. Está dividido em várias classes, algumas responsáveis pela manutenção de estruturas de dados essenciais ao bom funcionamento do simulador, tal como, classes mais front-end que fornecem uma boa experiência ao utilizador.