

# Relatório Football Manager

grupo 23:

Francisco Paiva, a93311

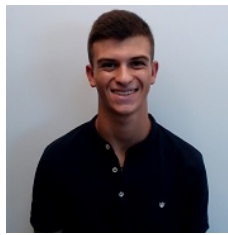
João Mendes, a93256

Ricardo Silva, a93195

12 de Junho de 2021



**Fig1: a93311**



**Fig2: a93256**



**Fig3: a93195**



**Universidade do Minho**  
Escola de Engenharia

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Jogador</b>	<b>3</b>
<b>3</b>	<b>Equipa</b>	<b>3</b>
<b>4</b>	<b>Jogo</b>	<b>4</b>
<b>5</b>	<b>Informacoes</b>	<b>5</b>
<b>6</b>	<b>Parser</b>	<b>5</b>
<b>7</b>	<b>Controller</b>	<b>5</b>
<b>8</b>	<b>View</b>	<b>6</b>
<b>9</b>	<b>FM</b>	<b>6</b>
9.1	Diagrama . . . . .	7
<b>10</b>	<b>Conclusão</b>	<b>8</b>

# 1 Introdução

No âmbito da unidade curricular Programação Orientada aos Objetos, o presente projeto consiste na realização do *Football Manager*. Desta forma, elaboramos varios objetos tais como: Jogador, Equipa, Jogo, Informacoes, Parser, Controller, View e Fm, com o intuito de respeitar a modularidade e torna-lo de fácil compreensão.

## 2 Jogador

Na Classe Jogador desenvolvemos as seguintes variáveis:

- nome - que se refere ao nome do jogador
- nCamisola - que se refere ao numero da camisola
- posicao - que indica a posição a que joga o jogador. Para sabermos a posição do jogador desenvolvemos um *enum*
- skills - que se refere as habilidades do jogador, elaborado através *Map*, onde a *key* que se refere às Habilidades e a *value* se refere à qualidade da habilidade
- historial - que se refere ao historial do jogador, através de uma *List* de equipas por onde o jogador jogou

Elaboramos os seus respetivos construtores bem como os seus *getters* e *setters* devido ao encapsulamento. Nesta classe, desenvolvemos um método que permite calcular o overall do jogador consoante a sua posição. Construímos ainda um método *Jogador parse* que recebe como argumento uma linha jogador do ficheiro logs e faz a conversão para o objeto.

Por fim, estão definidos as funções *toString*, *equals* e *clone*.

## 3 Equipa

Na Classe Equipa desenvolvemos as seguintes variáveis:

- nome - que se refere ao nome da equipa
- fundacaoEquipa - utilizamos um *LocalDate* para se referir a data de fundação da equipa

- jogadores - que se refere a lista de jogadores que cada equipa tem, utilizamos uma *List* de Jogador

Elaboramos os seus respetivos construtores bem como os seus *getters* e *setters* devido ao encapsulamento. Nesta classe, desenvolvemos um método que nos permite obter o jogador através do seu número da camisola *obterJogadorPeloNumero* e um método para calcular o *overall* da equipa. Para conseguirmos fazer a transferência de jogador de uma equipa para outra, tivemos de implementar dois métodos tais como *insereJogador* e *removeJogador*. Elaboramos também métodos que permitem converter para *string* a informação no ficheiro tais como o conteúdo da equipa, habilidades e posição.

Por fim, estão definidos as funções *toString*, *equals* e *clone*.

## 4 Jogo

Na Classe Jogo desenvolvemos as seguintes variáveis:

- tempo - que se refere à data do jogo
- estado - que se refere ao estado do jogo
- equipa1 - que se refere à equipa visitada
- equipa2 - que se refere à equipa visitante
- golosVisitada - que se refere aos golos da equipa visitada
- golosVisitante - que se refere aos golos da equipa visitante
- ld - que se refere a data que o jogo ocorreu
- jogadoresEquipa1 - que se refere aos titulares da equipa1
- jogadoresEquipa2 - que se refere aos titulares da equipa2
- taticaEquipa1 - que se refere a tática da equipa visitada
- taticaEquipa2 - que se refere a tática da equipa visitante
- substituiçoesEquipa1 - que se refere às substituições da equipa visitada
- substituiçoesEquipa2 - que se refere às substituições da equipa visitante

Elaboramos os seus respetivos construtores bem como os seus *getters* e *setters* com o devido encapsulamento. Nesta classe é onde é calculado o resultado de um jogo, sendo este processo dividido em dois métodos (*simulacao\_part1* e *simulacao\_part2*).

O resultado de um jogo é obtido através do overall de uma equipa e por um determinado número de iterações que tem como objetivo obter valores mais simpáticos. Além disso, estes métodos também tratam de fazer as substituições dos jogadores em campo, algo que é realizado no intervalo.

Para além disso, também é possível obter outras informações como a formação de determinada equipa bem como o resultado final ou as substituições que foram efetuadas.

## 5 Informacoes

Na Classe Informacoes desenvolvemos as seguintes variáveis:

- equipa - que se refere a todas as equipas
- jogadores - que se refere a todos os jogadores
- jogos - que se refere a todos os jogos

O objeto Informacoes tem como objetivo agrupar as diferentes informações num único objeto, assim conseguimos ter todos os jogadores, equipas e jogos num único objeto.

Para além disso, contém alguns métodos que são várias vezes usados ao longo do código para obter informações relativas a um jogador/equipa/jogo.

## 6 Parser

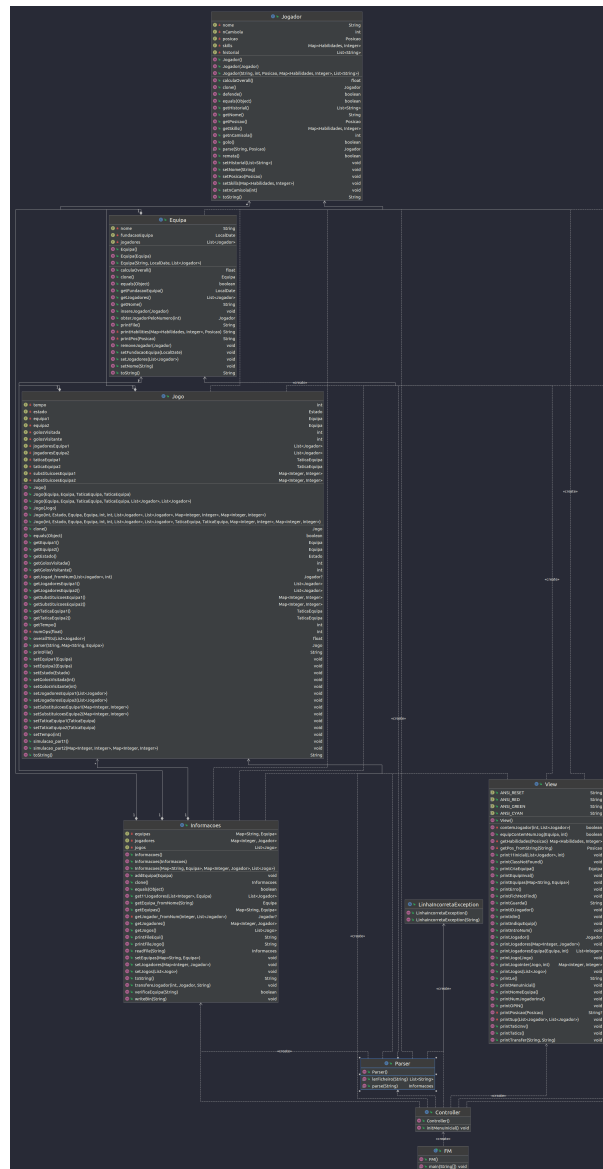
Elaboração desta classe permitiu a conversão da informação a partir da leitura do ficheiro.

## 7 Controller

Esta classe foi construída com o objetivo de estabelecer ligação entre a classe *View* com as restantes classes.



## 9.1 Diagrama



## 10 Conclusão

A realização deste projeto teve como objetivo por em prática a matéria lecionada nas aulas teóricas, bem como nas aulas práticas. Tivemos dificuldade na implementação de certos métodos para a realização da simulação do jogo, devido às várias abordagens que poderíamos ter.

O nosso jogo tem a capacidade de realizar as funcionalidades mais básicas mas também implementamos métodos mais complexos, bem como a simulação do resultado de um jogo podendo fazer alterações ao intervalo a ambas das equipas e essas alterações têm impacto no resultado final.

É possível também, criar um jogador e uma equipa de raiz, guardar e carregar estados e por fim aceder à informação que vai sendo guardada ao longo das ações do utilizador.

Em suma, estamos contentes com o nosso programa e com as suas funcionalidades, sentimos que estamos mais intrusados em *java* e que o progresso ao longo do semestre relativo à linguagem foi bastante positivo.