

[UNIVERSIDADE DO MINHO](https://www.uminho.pt/PT)

[Mestrado Integrado em Engenharia Informática](https://miei.di.uminho.pt/)

Programação Orientada Aos Objetos

Grupo 3

Carlos Daniel da Silva Santos (a85617)

Ana Paula Oliveira Henriques (a93268)

Ana Filipa Ribeiro Murta (a93284)

Ano Letivo 2020/2021

Índice

1. Introdução e principais desafios 3
2. Arquitetura de classes 4
   1. Teste . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   2. Jogador . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   3. Subclasses de Jogador . . . . . . . .. . . . . . . . . . . . . . . . . . . . 4
   4. CriarJogador . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
   5. Equipa . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
   6. CriarEquipa . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   7. EstadoJogo. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   8. Jogo . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   9. JogoRegisto . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   10. MainMenu . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   11. GerirMenu . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   12. JogMenu . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   13. Parser . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   14. ParserMod . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   15. Saver . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   16. Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   17. Exceptions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
3. Aplicação desenvolvida 10
4. Conclusão 10
5. Introdução e principais desafios

Este projeto consistiu no desenvolvimento de um jogo de futebol, semelhante ao conhecido jogo *Football Manager*, na linguagem de programação Java de forma a pôr em prática os conhecimentos aprendidos ao longo do semestre.

O programa desenvolvido consiste, portanto, numa partida entre duas equipas, uma identificada pela equipa que joga em casa e a outra pela equipa que joga fora, que foram escolhidas a partir de um ficheiro fornecido pelos docentes ou, então, criada pelo usuário.

Um jogo tem duração de 90 minutos em que, aos 45 minutos, é feito um intervalo para se efetuar substituições. Ao longo do jogo, serão feitas jogadas que serão determinadas consoante os números aleatórios gerados pela classe Random do pacote java.util.

O maior desafio enfrentado é capaz de ter sido as restrições dos jogadores consoante as suas posições ............................................................................................................................................................................................................................

**IMPORTANTE**: todas as classes estão associadas por composição.

1. Arquitetura de classes
   1. Teste

Esta classe é responsável por arrancar o programa, começando por imprimir no terminal o menu principal e, a partir daí, o usuário escolherá o que pretende fazer a seguir tendo em conta as opções que lhe são apresentadas.

* 1. Jogador

A cada tipo de jogador é associado um inteiro de forma a podermos pegar diretamente num objeto de tipo Jogador e saber se é um avançado, um médio, um lateral, um defesa ou um guarda-redes:

**private** static final int AVANCADO = 1;

**private** static final int MEDIO = 2;

**private** static final int LATERAL = 3;

**private** static final int DEFESA = 4;

**private** static final int GR = 5;

Um jogador tem um conjunto de características-base (cujos valores estão entre 0 a 100):

**private** String nome;

**private** int nr\_camisola;

**private** int velocidade;

**private** int resistencia;

**private** int destreza;

**private** int impulsao;

**private** int jogoCabeca;

**private** int remate;

**private** int capPasse;

**private** List<String> historico;

**private** int tipoJogador;

* 1. Subclasses de Jogador

A superclasse Jogador possui as seguintes subclasses:

Avancado:

**private** int drible;

Medio:

**private** int capRecuperacao;

**private** int dominioBola; -- característica extra

Lateral:

**private** int capCruzamento;

Defesa:

**private** int desarme;

GuardaRedes:

**private** int elasticidade;

**private** int lancamento; -- característica extra

Para além das características-base que nos foram fornecidas, foi-nos, também, proposto acrescentar características extra consoante o tipo do jogador. As únicas características extra que adicionámos foram: o lançamento de bola do guarda redes (com o pé ou com a mão) e o domínio da bola do médio, referente à sua capacidade de controlar a bola.

Cada jogador possui um valor de habilidade. Ora, os métodos que calculam a habilidade dum jogador encontram-se em cada uma das superclasses visto que este cálculo vai depender das características específicas de cada tipo de jogador. Quanto mais elevado for este valor, melhor é o desempenho esperado do jogador.

* 1. CriarJogador

Esta classe destina-se a criar um jogador com a intervenção do usuário, ou seja, é o usuário que escolhe que tipo de jogador é que pretende criar e lhe atribui um nome e valores às suas características. Depois disto, é perguntado ao utilizador a que equipa é que quer associar este novo jogador, inserindo-o nessa equipa caso esta exista.

* 1. Equipa