IPCA INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

**Licenciatura em Engenharia de Sistemas Informáticos**

Relatório de Trabalho Prático para

# Linguagens de Progamação II

André Cardoso & José Cosgrove

18848 & 18826

Trabalho realizado sob a orientação de:

Luís Ferreira

Barcelos, 26 de Abril de 2020

# Índice

# 1 Introdução

Foi proposto para a unidade curricular de Linguagens de Programação II, o desenvolvimento de uma solução em C♯ para problemas reais de complexidade moderada.
Para tal é necessário provar:

- Compreensão dos conceitos básicos de programação segundo o Paradigma Orientado a Objetos.
- Capacidade de Análise de problemas reais.

## 1.1 Motivação

Pretende-se o desenvolvimento de soluções em C♯ para problemas reais, trabalhando o paradigma de programação orientada a objetos.
Para esta Primeira Fase foi nos requerido uma implementação essencial das classes a utilizar, uma boa documentação do código, a criação do seu diagrama, a utilização de software gestor de versões e ainda a redação de um relatório de qualidade.

# 2 Enumeráveis

Para as classes **Arbitro** e **Jogador** é necessário indicar uma **ASSOCIACAO** e **CATEGORIA**, e uma **POSICAO**, respetivamente.

- **ASSOCIACAO**

  Contém todas as associações de Árbitros de Portugal.

- **CATEGORIA**

  Discriminação dos níveis profissionais que um árbitro pode alcançar.

- **POSICAO**

  Enumerável com as posições 'gerais' que um jogador pode ocupar no decorrer do jogo.

# 3 Classes

Para esta primeira fase foi-nos requerido uma implementação essencial. Assim sendo, as classes que vamos discutir incluem apenas os seus membros, construtores e propriedades; todas as classes possuem estas qualidades, e todos os membros têm a respetiva propriedade.

## 3.1 Pessoa

A classe **PESSOA** não contém todas as caraterísticas associáveis a um indivíduo mas apenas os membros básicos que consideramos importantes em contexto do problema: nome, data de nascimento, nacionalidade, altura e peso.

Todas as propriedades da classe possuem um get e um set sem qualquer tipo de tratamento ou avaliação, à exceção da **DataNascimento** que verifica se o valor atribuído é passível de atribuir à respetiva variável.

## 3.2 Jogador e Arbitro

Ambas as classes derivam da mesma classe base **PESSOA**, como tal estas herdam as variáveis e propriedades da classe base. No entanto, os construtores são únicos a cada classe diferente.

### 3.2.1 Jogador

Os membros únicos a esta classe são apenas *número* , *alcunha* e *posição* (**POSICAO**).

### 3.2.2 Arbitro

Nesta classe encontramos os membros *formação*(**DateTime**), *categoria*(**CATEGORIA**) e *associacao*(*ASSOCIACAO*).

## 3.3 Equipa, Jogo e Competição

Estas 3 classes são solitárias, não herdam nem são herança de nem para classe alguma. A única relação entre elas e as classes que já falamos é a existência de objetos gerados por elas. Nota: Nas próximas classes, será possível encontrar arrays, pelo que os construtores de cada classe contêm um ciclo que lhes permite acessar todas as posições, que é utilizado na construção do objeto.

### 3.3.1 Equipa

Membros:

**string** nome

**DateTime** fundacao

**Jogador[]** jogadores

**const int** max

**int** totJogadores

### 3.3.2 Jogo

Membros:

**Equipa** equipaA

**Equipa** equipaB

**Arbitro[]** arbitros

### 3.3.3 Competição

Membros:

**Equipa[]** equipas

**DateTime** inicio

**DateTime** fim

# Bibliografia