



UNIVERSIDADE D  
**COIMBRA**

## RELATÓRIO

Programação Orientada aos Objetos – POAOTrivia

André Lourenço Albuquerque - 2022231505  
Rodrigo Morenito Borges - 2022244993

# Índice

---

<b>Métodos e Classes</b> .....	<b>1</b>
Funcionamento de Métodos e Classes .....	2
<b>Manual de Utilizador</b> .....	<b>3</b>
<b>Conclusão</b> .....	<b>4</b>

## **Métodos e Classes:**

O nosso "POOTrivia" é constituído por 11 classes, sendo elas :

### **- Main:**

Onde criamos uma objeto instância da classe GUI, e utilizamos o método `guilnicial()`, de modo, a iniciarmos a interface

### **- Carrega:**

A classe Carrega carrega as perguntas a partir de um arquivo de texto, criando e colocando os dados dentro de uma lista de objetos (Perguntas). Têm ainda o método `escolhePerguntas()`, que faz com que apenas sejam perguntadas 5 perguntas aleatórias, independente de quantas estiverem dentro do ficheiro de texto.

### **- Perguntas:**

A classe Perguntas é uma superclasse que representa um pergunta genérica do jogo. Incluindo dentro dela atributos para armazenar todos os dados necessários para o quiz. Contém para além de métodos getters e setters, métodos como `modoHard`, `modoEasy` e `majoracao` que irão ser implementados nas suas subclasses, utilizando assim a herança

### **- Arte:**

A classe Arte estende a superclasse Perguntas, sendo que possui um construtor que chama o construtor da superclasse e inicializa atributos específicos. E utilizando a herança ajusta a quantidade de respostas dependo da dificuldade pretendida através dos métodos `modoHard` e `modoEasy`. Como também a respetiva majoração da subclasse.

### **- Ciências:**

A classe Ciências representa as perguntas do tipo Ciências, estendendo também a superclasse Perguntas, inicializando atributos específico como a lista de respostas difíceis. Tendo também os métodos `modoHard` e `modoEasy` ajustados para aparecer as opções fáceis antes da terceira pergunta e as difíceis depois. Tendo também a respetiva majoração da subclasse.

### **- Desporto:**

A classe Desporto representa perguntas do tipo Desporto e estende a superclasse Perguntas, inicializando os seus atributos específicos. Têm também o método `majoração Desporto` que irá ser utilizado nas subclasses de Desporto para calcular a pontuação da pergunta.

### **- Natação:**

A classe Natação estende a classe Desporto, inicializando atributos específicos para natação. O método `majoração` é sobrescrito para calcular a pontuação de cada pergunta de natação, utilizando também a `majoracaoDesporto` neste cálculo.

**- Futebol:**

A classe Futebol estende a classe Desporto, inicializando os seus respetivos atributos. O método majoração é sobrescrito para calcular a pontuação de cada pergunta, utilizando também a majoracaoDesporto neste cálculo. Utiliza ainda os métodos modoHard e modoEasy para mudar as perguntas consoante a dificuldade pretendida.

**- Ski:**

A classe estende a classe Desporto, inicializando atributos específicos para Ski. O método majoração é sobrescrito para calcular a pontuação de cada pergunta, utilizando a majoracaoDesporto neste cálculo.

**- registoJogo:**

A classe registoJogo representa o registo do desempenho do jogador durante o jogo e implementa a interface Serializabre para permitir criar um ficheiro de objetos, através do método guardarFicheiro, sendo este chamado para assim armazenar o progresso do jogador após o jogo.

**- GUI:**

A classe GUI representa a interface gráfica do jogo, gerenciando assim as perguntas, as opções de resposta, verifica as respostas do usuário, atualiza a pontuação, através de métodos como guilnicial, verificaResposta, guiFinal, reiniciarJogo

# Manual do Utilizador: "POOTrivia"

## 1. Introdução:

Bem-vindo ao "Quem Quer Ser Milionário". Este jogo desafia os teus conhecimentos num formato de perguntas e respostas. Este manual fornecerá orientações para participares nesta experiência.

## 2. Objetivo:

Acerta o maior número de perguntas possível respondendo corretamente a cinco perguntas de escolha múltipla.

## 4. Tipo de Perguntas:

Este jogo irá conter 5 perguntas de diferentes tipos : "Arte" , "Ciência" , "Desporto" , sendo as de desporto focadas em "Futebol" , "Ski" ou "Natação".

## 3. Responder Perguntas:

Cada pergunta pode ter uma dificuldade, fácil se for as primeiras perguntas ou difícil à medida que as perguntas vão avançando. Escolha a resposta correta entre as opções fornecidas, sendo algumas perguntas de Verdadeiro ou Falso e outras de escolha múltipla.

Ao longo de cada pergunta vais ter sempre no topo a pergunta , os botões com as possíveis opções onde podes selecionar a tua resposta de duas maneiras: Clicando com o rato em cima do botão, ou alternando as respostas com a tecla "Tab", e selecionar a resposta com a tecla "Espaço". Ao fim de seleccionares a resposta, aparecerá uma nova janela onde te informa se a tua resposta foi correta ou incorreta, e no primeiro caso apresenta os pontos que vais ganhar.

Cada resposta errada não retira pontos , porisso não tenhas medo de arriscar.

## 4. Fim de Jogo:

No fim do jogo irá te aparecer uma janela onde terás que escrever o teu nome e a data e hora em que fizeste o questionário no formato Ano/Mês/Dia, para o teu jogo ficar guardado. Ao fim disso o quiz termina e a janela fecha.

## Conclusão

Concluindo, conseguimos criar um programa que fizesse um quiz que desafia-se os conhecimentos em áreas distintas utilizando para isso bastantes conteúdos utilizados em sala de aula. Principalmente através dos conteúdos de herança e polimorfismo, mas também a implementação de JavaDoc e a criação de UML.