

1 2



9 0

FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
COIMBRA

# Relatório Projeto POO

Tiago Marques, Francisco Silva

Licenciatura em Engenharia Informática  
Faculdade de Ciências e Tecnologia da Universidade de Coimbra

9 de dezembro de 2023

# Conteúdo

<b>1</b>	<b>Resumo</b>	<b>2</b>
<b>2</b>	<b>Introdução</b>	<b>2</b>
<b>3</b>	<b>Descrição das classes e respetivos métodos</b>	<b>2</b>
3.1	Question . . . . .	2
3.2	Q_Art . . . . .	2
3.3	Q_Science . . . . .	3
3.4	Q_Sports . . . . .	3
3.5	Q_Ski . . . . .	3
3.6	Q_S_Football . . . . .	3
3.7	Q_S_Swim . . . . .	3
3.8	Game . . . . .	3
3.9	GUI . . . . .	3
3.10	POOTrivia . . . . .	4
<b>4</b>	<b>Estrutura dos Ficheiros</b>	<b>4</b>
4.1	Ficheiros de texto . . . . .	4
4.1.1	Perguntas . . . . .	4
4.1.2	Nomes dos ficheiros de texto . . . . .	4
4.2	Ficheiro de Objetos . . . . .	5
<b>5</b>	<b>Manual do Utilizador</b>	<b>5</b>
5.1	Menu . . . . .	5
5.1.1	Iniciar Jogo . . . . .	5
5.1.2	TOP3 . . . . .	7
5.1.3	Sair . . . . .	7
<b>6</b>	<b>Diagrama de Classes</b>	<b>7</b>
<b>7</b>	<b>Javadoc</b>	<b>8</b>
<b>8</b>	<b>Conclusão</b>	<b>8</b>

# 1 Resumo

De um modo geral, foi-nos proposto o desenvolvimento de um jogo de perguntas e respostas. O jogo será implementado na linguagem Java e este trabalho abordará os seguintes aspetos:

- Elaboração de um diagrama de classes (UML);
- Elaboração do código da aplicação em Java;
- Elaboração do Javadoc

## 2 Introdução

O presente relatório foi desenvolvido no âmbito da unidade curricular de Programação Orientada Aos Objetos em que nos foi proposto a realização de um trabalho prático que pretende desenvolver um jogo de perguntas e respostas. Visto isto, desenvolvemos uma aplicação, para gerir o mesmo e em que a interação com o utilizador se efetuará através de uma interface gráfica, ou seja, vai aparecendo um menu com várias opções em que o utilizador escolherá o que pretende fazer ao longo da aplicação. Este relatório serve de base para consolidar a matéria dada.

## 3 Descrição das classes e respetivos métodos

A aplicação desenvolvida apresenta diversas classes e métodos importantes. Sendo desenvolvida em Java, a relação entre estas encontra-se apresentada de forma mais explícita ao nível do UML associado a este documento. De seguida, será apresentada uma breve explicação de cada uma das classes.

### 3.1 Question

A classe ***Question*** é uma classe abstrata que serve como um modelo genérico para representar as perguntas no jogo. Ela encapsula atributos e comportamentos comuns a diferentes tipos de perguntas, permitindo a extensão para tipos específicos de perguntas por meio de subclasses.

### 3.2 Q\_Art

A classe ***Q\_Art*** é uma extensão da classe *Question* e representa um tipo específico de pergunta relacionada à arte. Ela herda os atributos e comportamentos da classe *Question* e adiciona funcionalidades específicas para lidar com perguntas de arte (respostas fáceis e difíceis).

### 3.3 Q\_Science

A classe ***Q\_Science*** é uma extensão da classe *Question* e representa um tipo específico de pergunta relacionada à ciência. Ela herda os atributos e comportamentos da classe *Question* e adiciona funcionalidades específicas para lidar com perguntas de ciência.

### 3.4 Q\_Sports

A classe ***Q\_Sport*** é uma classe abstrata que estende a classe *Question*, servindo de molde para representar um tipo específico de pergunta relacionada aos desportos existentes (Futebol, Ski e Natação).

### 3.5 Q\_Ski

A classe ***Q\_S\_Ski*** é uma subclasse de *Q\_Sport*, especificamente para representar perguntas relacionadas ao desporto de Ski. Ela herda a estrutura básica de uma pergunta esportiva da classe abstrata *Q\_Sport* e fornece comportamentos específicos para questões de esqui.

### 3.6 Q\_S\_Football

A classe ***Q\_S\_Football*** é uma subclasse de *Q\_Sport*, projetada especificamente para representar perguntas relacionadas ao futebol. Ela herda a estrutura básica de uma pergunta desportiva da classe abstrata *Q\_Sport* e fornece comportamentos específicos para questões de futebol (respostas em formato de Camisola e de Nome).

### 3.7 Q\_S\_Swim

A classe ***Q\_S\_Swim*** é uma subclasse de *Q\_Sport*, destinada a representar perguntas relacionadas à natação. Ela herda a estrutura básica de uma pergunta desportiva da classe abstrata *Q\_Sport* e fornece comportamentos específicos para as questões de natação.

### 3.8 Game

A classe ***Game*** é responsável por gerenciar um jogo, armazenando informações sobre o jogo, como a lista das perguntas, respostas corretas, estágio atual do jogo, nome do jogador, data e hora do jogo. Para além disso, é dentro desta classe que acontece todo o gerenciamento de ficheiros, tanto de objetos como de texto.

### 3.9 GUI

A classe ***GUI*** é onde a interface gráfica é criada e permite ao utilizador que este tenha uma melhor experiência do do programa.

### 3.10 POOTrivia

A classe ***POOTrivia*** é onde se encontra o método *Main*. A criação de uma classe própria para o *Main* serve para proteger o utilizador do código e proteger o código do utilizador.

## 4 Estrutura dos Ficheiros

### 4.1 Ficheiros de texto

Para este projeto são usados dois ficheiros de texto:

#### 4.1.1 Perguntas

O processo de leitura do ficheiro de perguntas foi efetuado através do uso da função de separar cada parcela correspondente a cada atributo de cada classe por uma barra ("/"). Desta forma, será retornado um vetor de String em que cada posição será associada ao respetivo atributo da classe de modo a permitir a fácil e rápida edição do ficheiro. O ficheiro ficará no formato:

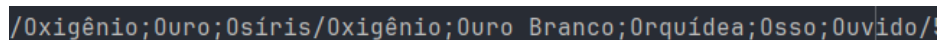
*Categoria/Pergunta/Opção Correta;Opção1;Opção2;...;OpçãoN/Quantidade De Pontos*

Após a leitura de cada linha do ficheiro, a parte correspondente às opções é processada adicionalmente. Essa secção, após a separação inicial por barra ("/"), é posteriormente dividida usando o ponto e vírgula(";") como separador. Isso resulta na obtenção de uma lista de opções individuais para a pergunta. Contudo há alguns casos especiais:

- No caso do **Futebol**, ainda é feita mais uma subdivisão utilizando o *underscore* ("\_"), guardando, em listas diferentes, o nome do jogador e o seu respetivo número.

*OpçãoCorretaNomes\_OpçãoCorretaCamisola;...;OpçãoNNomes\_OpçãoNCamisola*

- No caso das **Ciências**, na divisão das opções há mais uma subdivisão a fazer através da barra ("/"), sendo que há uma lista de opções fáceis e outra de opções difíceis. A primeira parcela da subdivisão corresponde à lista de opções fáceis e a outra subdivisão, à lista de opções difíceis.



```
/Oxigênio;Ouro;Osiris/Oxigênio;Ouro Branco;Orquídea;Osso;Ouvindo/;
```

Figura 1: Exemplo das parcelas das respostas em "Ciências"

- No caso de **Ski** ou **Natação**, como ambos têm as respostas "Verdadeiro" ou "Falso" no ficheiro apenas se coloca a opção correta

*Categoria/Pergunta/Opção Correta/Quantidade De Pontos*

#### 4.1.2 Nomes dos ficheiros de texto

Um ficheiro de texto foi criado para armazenar os nomes dos ficheiros de objetos associados aos jogos previamente jogados. Cada linha desse ficheiro contém o nome de um ficheiro específico, correspondente a um ficheiro de objeto individual.

## 4.2 Ficheiro de Objetos

À medida que o utilizador joga, esses jogos são salvos em um arquivo de objetos, posteriormente utilizado para calcular o TOP 3 dos melhores jogos guardados em memória.

# 5 Manual do Utilizador

O programa interage com o utilizador através de uma interface gráfica. O nosso objetivo desde o início foi criar uma interface gráfica intuitiva para permitir que o utilizador possa fazer as ações que pretende sem ter qualquer tipo de dúvida no decorrer do jogo.

## 5.1 Menu

Quando o jogo é iniciado, é-lhe apresentado um Frame do menu que contém 3 opções na qual o utilizador escolhe para onde quer prosseguir.

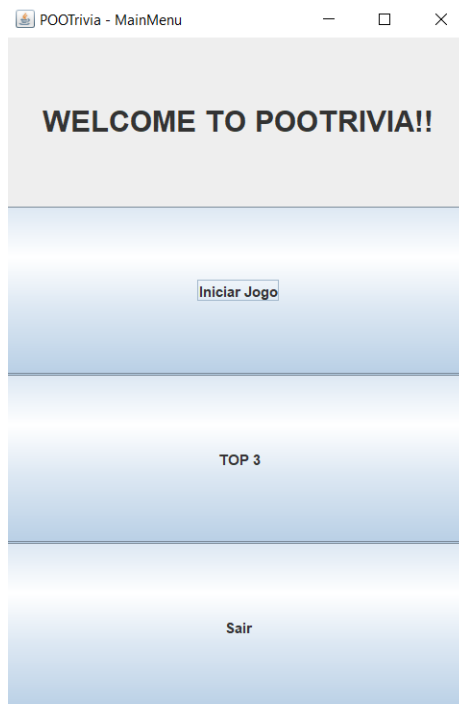


Figura 2: Menu do Jogo

### 5.1.1 Iniciar Jogo

Logo após pressionar este botão o jogo inicia-se. O Frame fica maior e aparece a primeira pergunta. Na parte superior aparece a categoria da pergunta. Logo abaixo, o seu enunciado e por fim na parte mais inferior do Frame, botões com as respostas.

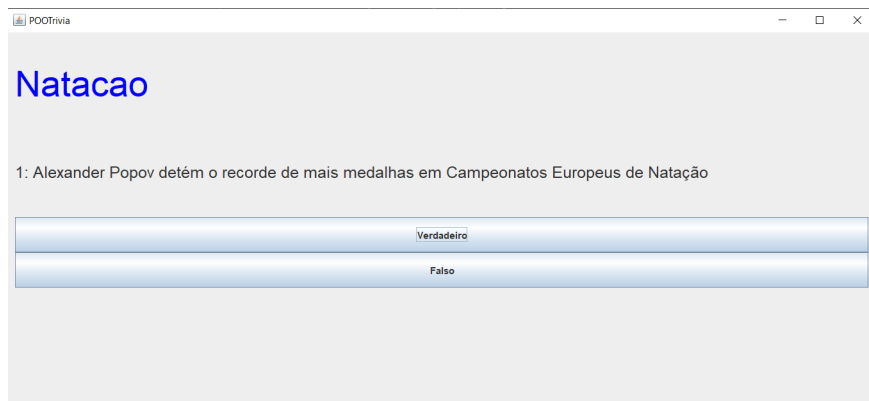


Figura 3: Exemplo de Pergunta

Quando o utilizador clica num dos botões, haverá duas opções:

- Utilizador carregou na **resposta certa**
- Utilizador carregou na **resposta errada**

Se a **resposta correta** for seleccionada, duas caixas de mensagem serão exibidas: uma indicando que a resposta está correta e outra mostrando a sua pontuação naquela pergunta.



Figura 4: Mensagens para a resposta correta

Se a **resposta errada** for seleccionada, duas caixas de mensagem serão exibidas: uma indicando que a resposta está errada e outra mostrando a opção correta daquela pergunta.



Figura 5: Mensagens para a resposta errada

Já no final do jogo, aparecerá uma caixa de texto para introduzir o nome do utilizador e uma mensagem com a pontuação final do utilizador.

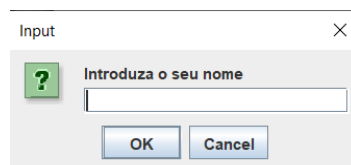


Figura 6: Caixa de texto para introduzir o nome

### 5.1.2 TOP3

Ao pressionar o botão do **TOP3**, uma nova janela é aberta onde apresenta as 3 maiores classificações de jogos anteriores. Nelas, está presente o nome do utilizador, a data, a hora e a sua respetiva classificação.



Figura 7: Exemplo de janela do TOP3

### 5.1.3 Sair

O botão de **Sair**, serve para sair do jogo. Ao ser pressionado, uma mensagem aparece perguntando se o utilizador deseja sair. Caso se confirme, o programa fecha.

## 6 Diagrama de Classes

O Diagrama de Classes, composto pelas classes do programa e pela relação entre elas. Este dá um panorama geral do programa e encontra-se apresentado de seguida. No entanto, com o intuito de ser mais perceptível, será disponibilizado em PDF com os restantes elementos do trabalho.

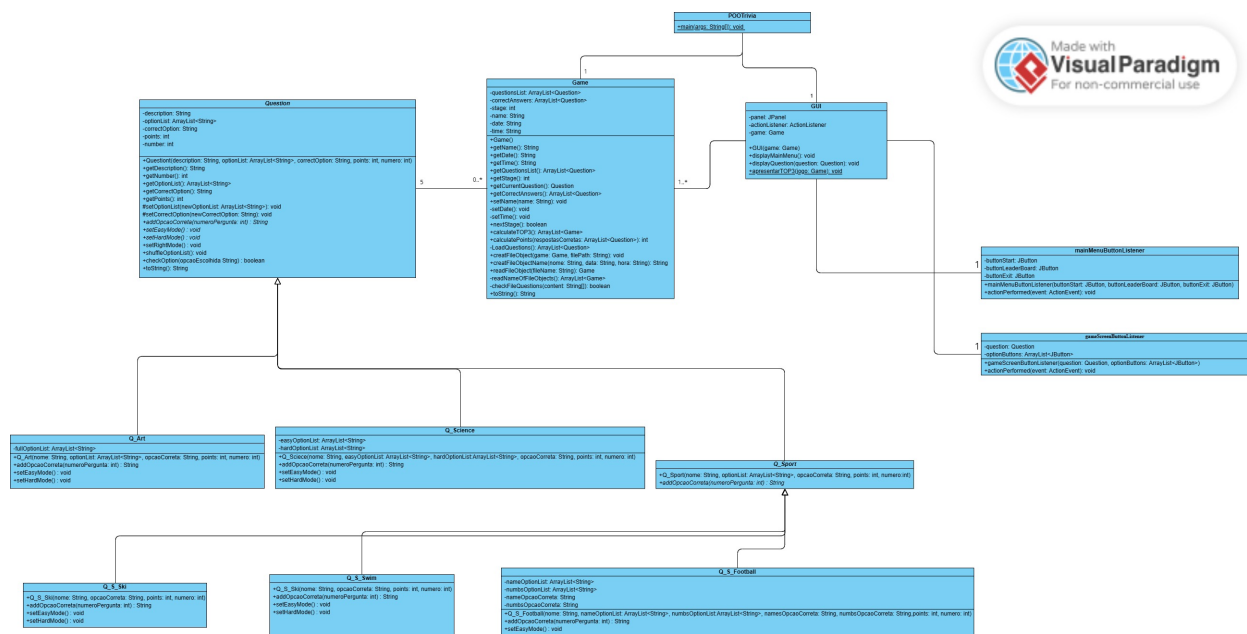


Figura 8: Diagrama de Classes do Jogo



## 7 Javadoc

Como já é de conhecimento geral, o Javadoc é uma ferramenta utilizada para gerar documentação, em formato HTML, para funções produzidas em Java. Desta forma, as páginas em HTML, são produzidas a partir dos comentários introduzidos no código fonte, no formato Javadoc. Este formato de comentários apresenta uma forma específica de ser efetuada, ou seja, é necessário escrever “/\*\*” e de seguida clicar em *Enter*. Desta forma, será iniciado um comentário em Javadoc, normalmente localizado antes do início de cada classe e antes do início de cada método.

## 8 Conclusão

Ao longo da realização deste trabalho prático foi possível consolidar e pôr em prática diversos conceitos abordados nas aulas Teórico-práticas da cadeira de Programação Orientada aos Objetos, relacionados com herança, polimorfismo e ficheiros, manipulando de forma eficaz os objetos. É importante referir a importância deste projeto no que toca à interface gráfica, visto que este interligou os conceitos abordados nas aulas com o mundo cibernético. Sendo assim, as noções desta cadeira estão, agora, mais claras.