

# RELATÓRIO DE PROJETO

PROGRAMAÇÃO EM JAVA

POAO TRIVIA

**Unidade Curricular:**

Programação Orientada aos Objetos

**Realizado por:**

Luana Carolina Reis

Nº 2022220606

# ÍNDICE

## 1. Introdução

- 1.1. Objetivos
- 1.2. Cuidados a ter

## 2. Classes, métodos e funcionamento

- 2.1. Classes
- 2.2. Métodos
- 2.2. Funcionamento

## 3. Manual do utilizador

- 3.1. Tipologia e dificuldade de perguntas
- 3.2. Majoração da pontuação
- 3.3. Funcionalidades disponíveis

## 4. Conclusão

## 5. Referências



# INTRODUÇÃO

## OBJETIVOS

O "POAO Trivia" é um projeto escrito em JAVA, que consiste num jogo de perguntas e respostas. O objetivo é testar os conhecimentos do jogador em três temas diferentes: Artes, Desporto (Futebol, Natação e Ski) e Ciências.

Cada jogo é composto por cinco perguntas, escolhidas aleatoriamente e sem repetição e as opções de resposta disponíveis estão dependentes do avançar do jogo, com consequente aumento da dificuldade do mesmo.

Por cada pergunta correta, é obtida uma pontuação, calculada com base num sistema de majoração, que utiliza a tipologia de pergunta como critério de avaliação.

No fim de cada rodada, o utilizador terá acesso a um "ranking" de pontuações, em que é exibido um TOP 3 e a pontuação de jogo atual. Após a sua visualização, terá ainda a possibilidade de reiniciar o jogo.

Tudo isto só será possível de atingir com recurso a uma interface gráfica JFrame, que permite a utilização de botões, caixas de texto e outros elementos visuais, que tornam a jogabilidade mais intuitiva e "user friendly".

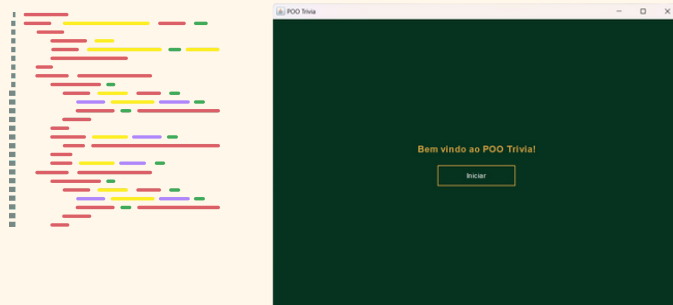


Fig. 1 - Painel de início de jogo

## CUIDADOS A TER

Ao longo do desenvolvimento deste projeto, foram seguidos os princípios da programação orientada aos objetos, com a criação de classes adequadas ao problema, proteção de variáveis e métodos, utilização de herança e polimorfismo, remoção de redundâncias no código e respeito pelas boas práticas de programação. Desta forma, foi possível atingir uma melhor organização do código, que o torna mais eficiente e mais prático de manter e atualizar.

Outro aspecto a ter em conta é a utilização de recursos de tratamento de exceções, que permitem lidar com erros e situações inesperadas, causados pela manipulação de ficheiros e de dados inseridos pelo utilizador. Isto garantiu que o jogo funcionasse de forma estável, mesmo em condições adversas.



# CLASSES, MÉTODOS E FUNCIONAMENTO

## CLASSES



**Classe “Trivia”:** Classe gestora principal, responsável pela leitura de perguntas a partir do ficheiro “pootrivia.txt” e chamada da GUI.

**Classe “GUI”:** Através de um “CardLayout”, são mostrados os diversos painéis consecutivos, de início de jogo, perguntas, fim de jogo e TOP 3. É nesta classe que está a lógica fundamental para a correta execução do programa, com implementações de “listeners” de botão e de rato.

**Classe “Escolha”:** Responsável por escolher cinco novas perguntas sempre que o jogo é carregado pela primeira vez ou reiniciado pelo utilizador.

**Classe “Ficheiros”:** Classe gestora de ficheiros, responsável por abrir, fechar, ler e escrever em ficheiros “.txt” e “.dat”.

**Classe “Resultados”:** Escreve os resultados de um jogo (data e hora, nome, perguntas certas e perguntas erradas) num ficheiro objeto.

**Classe “Top”:** Envia todos os ficheiros objeto para uma lista temporária, que é ordenada de forma a obter os três melhores jogos e mostrá-los ao utilizador.

**Classes “Pergunta”, “Artes”, “Ciências”, “Desporto”, “Futebol”, “Natação” e “Ski”:** Classes em que são aplicados os conceitos de herança. Consoante o ponto do jogo em que o jogador se encontra, as opções de resposta serão diferentes e obtidas através destas classes, bem como a pontuação de perguntas, que segue regras específicas para cada subclasse.

# CLASSES, MÉTODOS E FUNCIONAMENTO

## MÉTODOS

Nesta secção, são explicados os métodos de maior importância para o funcionamento do jogo.

- **getEasyAnswers():** É chamado para a 1.<sup>a</sup> e 2.<sup>a</sup> questão e devolve uma lista com as opções de resposta fáceis. No caso das perguntas de "Arte", será necessário criar uma sublista de tamanho 3, que terá obrigatoriamente que conter a resposta correta. Nas perguntas de "Ciências" e "Futebol", há uma lista de respostas fáceis, de acesso direto no ficheiro de texto.
- **getHardAnswers():** É chamado para as restantes questões do jogo e devolve uma lista que contém as opções de resposta difíceis. Para "Artes", "Ciências" e "Futebol", as opções são também de acesso direto no ficheiro.
- **getAnswers():** As perguntas de tipo "Natação" e "Ski" não são afetadas pelo avançar do jogo, sendo apenas de Verdadeiro/Falso.
- **getScore():** Cálculo da pontuação de cada pergunta, de acordo com a tabela de majoração.
- **handleAnswer():** Compara o texto da opção escolhida com o resultado presente no final de cada linha do ficheiro de texto. Se for igual, adiciona a uma lista de perguntas acertadas. Caso contrário, adiciona a uma lista de perguntas erradas.
- **getTopScores():** No fim do jogo, é necessário calcular os três melhores jogos. Para isto, são guardados todos os jogos numa lista temporária e, para cada um deles, é feita a leitura das linhas com as perguntas certas, ao mesmo tempo que se calcula a pontuação de jogo. Depois disto, os jogos serão organizados com recurso a um método de organização `Arrays.sort()`, sendo selecionados os três primeiros índices (se existirem) e devolvendo-se essa informação para a GUI.

## FUNCIONAMENTO

O jogo segue um fluxo de funcionamento através de diversos painéis, cujo objetivo é aumentar a qualidade da "interface" gráfica e melhorar a experiência do utilizador. Esta passagem entre painéis é feita de forma automática, assim que é detectada uma ação. Neste caso, essa ação será um clique num botão `JButton`. Para cada pergunta, haverá um número variável de botões disponíveis para resposta. Após a seleção de um deles, são chamados os métodos que vão verificar se o jogador acertou ou errou a pergunta e, de seguida, ser-lhe-á mostrado o resultado, por 500 milissegundos, antes de passar para a seguinte. Este processo continua até chegar à última, em que é pedido o preenchimento do campo de nome, para efeitos de posterior armazenamento de informações de jogo. Isto será fundamental para depois se poder visualizar o TOP 3.

# MANUAL DO UTILIZADOR

## TIPOLOGIA E DIFICULDADE DE PERGUNTAS

As perguntas podem ser dos seguintes tipos:

- Artes (5 ou 3 opções)
- Ciências (respostas fáceis ou difíceis)
- Futebol (nomes ou números de jogadores)
- Natação (verdadeiro ou falso)
- Ski (verdadeiro ou falso)



Na 1.<sup>a</sup> e 2.<sup>a</sup> perguntas do jogo, são consideradas as opções de resposta FÁCIL.

Na 3.<sup>a</sup>, 4.<sup>a</sup> e 5.<sup>a</sup> perguntas do jogo, são consideradas as opções de resposta DIFÍCIL.

## MAJORAÇÃO DA PONTUAÇÃO

A pontuação de pergunta é obtida através de uma pontuação base de 5\_PONTOS, majorada de acordo com a seguinte tabela.

Área	Características	Majoração da pontuação
Artes	-	x 10
Ciências	-	+ 5
Desporto	-	+ 3
Futebol	Subárea de Desporto.	+ 1
Natação	Subárea de Desporto.	+ 10
Ski	Subárea de Desporto.	x 2

## FUNCIONALIDADES DISPONÍVEIS

Como extras, foram implementadas ainda algumas funcionalidades de acessibilidade:

- Seleção de botões com "Enter"
- Foco visual num botão, quando o rato passa por cima
- Automatização na passagem de perguntas, com recurso a um timer
- Proteção contra nomes inválidos (só podem conter letras e espaços)

»»»»»»»» FIM DE JOGO ««««««««

Nome do jogador:



# CONCLUSÃO

## CONCLUSÃO

Com a realização deste projeto, foram fortalecidas capacidades de raciocínio lógico e de organização, pela necessidade de trabalhar com múltiplas classes interligadas entre si e por ter que fazer tudo isto trabalhar juntamente com a JFrame.

Para além de ficar a conhecer melhor as potencialidades da linguagem Java, foram obtidos conhecimentos fora do âmbito da disciplina.



# REFERÊNCIAS

[ 1 ] Oracle, "Java Documentation". [online]. Disponível em: <https://docs.oracle.com/en/java/>. [Consultado em 17/11/2023].

[ 2 ] StackOverflow, "Allowing the "Enter" key to press the submit button, as opposed to only using MouseClick". [online]. Disponível em: <https://stackoverflow.com/questions/13731710/allowing-the-enter-key-to-press-the-submit-button-as-opposed-to-only-using-mo>. [Consultado em 08/12/2023].

[ 3 ] W3Schools, "Java Files". [online]. Disponível em: [https://www.w3schools.com/java/java\\_files.asp](https://www.w3schools.com/java/java_files.asp). [Consultado em 08/12/2023].

