**Faculdade de Ciências e Tecnologia da Universidade de Coimbra**

**Departamento de Engenharia Informática Licenciatura Engenharia Informática**

****

Programação Orientada aos Objetos 2022/2023 Projeto

POOTrivia – Jogo de Questões

Bruno Almeida Nº: 2021237081

Rui Ribeiro Nº: 2021189478

# Índice

* Introdução\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3
* Organização de dados\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3
* Classes\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3-5
* Conclusão\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5
* Bibliografia\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

# Introdução

Com o objetivo de planear e organizar um questionário criamos uma aplicação que permitisse a um utilizador resolver um questionário e guardar a sua pontuação. Como tal, decidimos montar uma estrutura eficiente e que pudesse ser útil. Inicialmente queríamos algo simples, como uma interface que tivesse algumas funcionalidades como criar um novo questionário e listar o Top3 de todos os jogos guardados na base de dados.

## Organização de dados

De modo a facilitar a organização dos dados, organizamos a informação por arrayLists específicas para cada categoria, onde apenas existem informações de cada tipo. Isto dava-nos uma grande facilidade no acesso aos dados de questionários.

Vamos buscar informações sobre o ficheiro de perguntas.txt e atribuímos-lhe um arrayLists. Para um melhor acesso ao ficheiro de texto, a informação está dividida por “/”, onde a primeira informação é o tipo de questão, seguida pela questão, as opções de resposta(estas encontram-se divididas por “;”) e por fim a opção correta.

Criámos também diversas classes, para ser mais otimizado.

# Classes

## Classe Abstrata Pergunta

A pergunta é caracterizada pelo seu tipo(Desporto, Artes, Ciências), o seu texto, a pontuação “pont”, pelas opções de resposta “options” e por fim a opção correta “opCorreta.

## Classe Desporto

A classe chamada "desporto" estende a classe "pergunta". A classe possui atributos como "subtipo" e "major" para fornecer especificidade adicional e pontuação para perguntas esportivas. Na classe existem métodos para recuperar o subtipo e calcular a pontuação final com base na majoração, nos pontos da pergunta e nos subtipos específicos.

## Classe Ciências

A classe chamada "ciencias", assim como no exemplo anterior, esta classe estende a classe "pergunta" e inclui atributos como "major" para majoração, melhorando o mecanismo de pontuação para perguntas de ciências. Além disso, há um método para calcular a pontuação final com base na majoração e nos pontos da pergunta.

## Classe Artes

A classe 'artes', representa uma pergunta relacionada às artes. Semelhante aos exemplos anteriores, esta classe estende a classe 'pergunta' e incorpora atributos como 'major' para majoração. Além disso, há um método para calcular a pontuação final com base na majoração e nos pontos da pergunta.

**Sobre a aplicação(interfaceJogo):**

Criamos uma classe chamada **interfaceJogo**, onde se encontra a nossa *main*. Ela começa logo por carregar as perguntas no ficheiro “perguntas.text”, inicializa o gerenciador de jogos e por fim inicia a parte gráfica.

Na classe estão presentes vários métodos como é o caso de: “carregarPerguntas”, “exibirMenuPrincipal”, “iniciarNovoJogo” e “exibirTop3”.

**carregarPerguntas –** esta função recebe como parâmetro o path de onde se encontra as perguntas que lhe iram servir de data base e devolve uma lista do tipo perguntas.

**exibirMenuPrincipal –** esta recebe a lista de perguntas obtida pelo método anterior, de seguida a função exibe no ecrã o menu principal, de onde saíram todas as restantes opções de interação, com o GUI. Nele estão presentes dois botões, o novo jogo e exibir o top 3 jogos.

**iniciarNovoJogo –** esta função recebe a lista de perguntas da função *carregaPerguntas* como o nome do jogador. Ela chama então a gerenciarJogos, que irá tratar dos questionários.

**exibirTop3 –** este método apenas chama a gerenciarJogos para mostrar o top 3 de jogos com maior pontuação.

**Sobre a aplicação(gerenciadorJogos):**

Criamos uma classe chamada **gerenciadorJogos**, que irá tratar da parte dos jogos e de chamar a classe **jogo,** responsável por apresentar os questionários. Nesta classe estão presentes funções como: “carregarJogos”, “salvarJogos”, “exibirTop3” e “lerArquivo”.

**carregarJogos –** neste método são carregados todos os jogos presente no ficheiro “.dat”, logo de seguida adiciona os jogos à lista jogos, e ordena-a de maior pontuação para menor, e devolve a lista ordenada.

**salvarJogos –** aqui cria o diretório jogo (onde os .dat são guardados) caso já não exista. Gera o arquivo com base na formatação fornecida no enunciado e preenche os dados dessa forma.

**exibirTop3 –** esta função ordena a lista "jogados" com base nos valores de "Pontuacao" em ordem decrescente, adiciona à lista top3List os 3 jogos com maior pontuação, e por fim imprime na consola.

**lerArquivo –** Cria o arquivo, onde serão guardados os jogos.

**Sobre a aplicação(jogo):**

Criamos uma classe chamada **jogo**, que irá tratar da parte de receber as perguntas e o nome do jogador e iniciar o jogo e tratas das perguntas e das questoes que vao ser apresentadas. Nesta classe estão presentes funções como: “jogar”, “getRandomPergunta”, “exibirPergunta” , “getRandomOps” e “getUserChoice .

**jogar –** este é o metodo responsavel pelo jogo em si.

**getRandomPergunta –** este metodo é responsavel por dar shuffle a todas as perguntas e selecionar uma random

**exibirPergunta –** este metodo exibe a pergunta escolhida ao utilizador

**getRandomOps –** este metodo retorna opções de resposta random de acordo com as opções existentes.

**GetUserChoice –** por fim este devolve a escolha de resposta do utilizador

## Conclusão

Com a conclusão deste projeto, alcançamos um avanço significativo em nossas habilidades de programação, aprimorando nossa capacidade de construir código de maneira eficaz. Evoluímos também na forma como organizamos código. Este projeto serviu como uma oportunidade valiosa para aprender, crescer e consolidar nosso conhecimento prático na construção de software.

# Bibliografia

* <https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>
* Slides de Apoio (dadas nas aulas práticas)