

## **Sistema de Gerenciamento e Avaliação por Quiz com JavaFX e Hibernate**

**Aluno:** Lucas Lucena

**Professor:** Prof. Sandino

---

# **1. Introdução**

Este projeto implementa um sistema completo de gerenciamento e avaliação por meio de quizzes, desenvolvido com JavaFX para a interface gráfica e Hibernate para persistência de dados em banco de dados.

O sistema permite o cadastro e gerenciamento de usuários, criação de questões, aplicação de quizzes com controle de tempo, e apresenta um ranking de pontuação.

# **2. Funcionalidades**

- Login de usuário e administrador
- Cadastro de novos usuários
- Cadastro e edição de questões
- Escolha de dificuldade (Fácil, Médio, Difícil)
- Realização de quizzes com temporizador
- Pontuação baseada na dificuldade
- Exibição de ranking geral
- Gerenciamento de usuários (exclusão de usuários comuns)
- Interface responsiva e intuitiva

# **3. Fluxo Geral do Sistema**

1. O usuário realiza o login.
2. Dependendo do tipo de conta:

- **Usuário comum:** Escolhe dificuldade e inicia o quiz ou visualiza o ranking.
  - **Administrador:** Pode cadastrar questões, editar/excluir questões, gerenciar usuários e visualizar o ranking.
3. Durante o quiz:
- O usuário responde as questões.
  - O tempo é controlado (60 segundos).
  - A pontuação é atribuída com base na dificuldade.
4. Após o quiz:
- O resultado é mostrado.
  - A pontuação é salva no ranking.

## 4. Estrutura de Métodos

- `start(Stage primaryStage)`: Inicializa a aplicação.
- `criarLoginPane()`: Cria o painel de login.
- `criarCadastroPane(VBox loginPane)`: Cria o painel de cadastro.
- `alternarPanels(Node mostrar, Node esconder)`: Alterna painéis.
- `iniciarQuiz(String dificuldade, String usuario)`: Inicia o quiz.
- `finalizarQuiz(Stage quizStage, int acertos, int pontos, List<Questao> questoes, String usuario)`: Finaliza o quiz.
- `abrirRankingAdmin()`: Visualiza e gerencia o ranking como admin.
- `abrirRankingUser()`: Visualiza o ranking como usuário.
- `abrirPainelUsuario()`: Tela principal para usuários comuns.
- `abrirPainelAdmin()`: Tela principal para administradores.

- `abrirCadastroQuestoes()`: Cadastro de novas questões.
- `cadastrarUsuario(String nome, String senha, boolean admin)`: Cadastro de novos usuários.
- `abrirVisualizarQuestoes()`: Visualiza todas as questões.
- `abrirCadastroQuestoesEdicao(Questao questao)`: Edita uma questão existente.
- `abrirGerenciarUsuarios()`: Gerencia e exclui usuários.
- `showAlert(Alert.AlertType type, String title, String message)`: Exibe alertas.

## 5. Manual de Uso

### 5.1 Abertura do Sistema

- Tela inicial de Login.
- Opções: Login ou Cadastro de novo usuário.

### 5.2 Cadastro de Novo Usuário

- Clique em "Cadastre-se".
- Preencha Nome, Senha e escolha se será Admin.
- Cadastre e retorne à tela de login.

### 5.3 Login

- Insira Nome e Senha.
- Direcionamento automático para o painel adequado.

## **5.4 Painel de Usuário**

- Escolha dificuldade e inicie o quiz.
- Opcionalmente, visualize o ranking.

## **5.5 Painel de Administrador**

- Cadastrar Questões.
- Visualizar/Editar/Excluir Questões.
- Gerenciar usuários.
- Visualizar e gerenciar o ranking.
- Logout ou sair do sistema.

### **5.5.1 Cadastro e Edição de Questões**

- Preencha Enunciado, Alternativas, Correta e Dificuldade.
- Salve para registrar ou atualizar questões.

### **5.5.2 Gerenciamento de Usuários**

- Visualize usuários comuns.
- Exclua se necessário.

### **5.5.3 Ranking de Pontuação**

- Visualização de pontuações.
- Exclusão de registros apenas por administradores.

## **6. Conclusão**

O sistema de quiz apresentado oferece uma solução robusta para avaliações interativas, combinando controle de usuários, pontuações inteligentes, e uma interface amigável.

Foi desenvolvido para ser fácil de utilizar, extensível e modular, podendo ser adaptado para diferentes contextos educacionais e profissionais.