**INSTITUTO MAUÁ DE TECNOLOGIA**

**ENGENHARIA DA COMPUTAÇÃO**

FELIPE FAZIO DA COSTA; RA: 23.00055-4

FERNANDO GODOI GRINEVICIUS; RA: 22-00832-2

MATHEUS ANTÔNIO DA LUZ CARDOSO; 22.01059-9

RUTH RAMOS ROMEU; RA: 22.01003-3

JONAS FERNANDO DA SILVA EBOLI MACHADO; 22.0

**Aktooh**

Projeto Semestral de linguagens de programação – 1° Semestre

SÃO CAETANO DO SUL

2025

Sumário

[1. INTRODUÇÃO 3](#_Toc199751770)

[2. FUNDAMENTAÇÃO TEÓRICA 4](#_Toc199751771)

[3. METODOLOGIA 5](#_Toc199751772)

[4. DESENVOLVIMENTO DO SISTEMA 7](#_Toc199751773)

[4.1 Estrutura Geral do Sistema 7](#_Toc199751774)

[4.2 Funcionalidades Implementadas 9](#_Toc199751775)

[4.3 Integração com Banco de Dados 10](#_Toc199751776)

[5. RESULTADOS E DISCUÇÃO 15](#_Toc199751777)

## 1. INTRODUÇÃO

Este relatório tem como objetivo descrever o desenvolvimento de um sistema de quiz interativo, inspirado na dinâmica do aplicativo Kahoot, utilizando a linguagem de programação Java e integrando um banco de dados MySQL para o armazenamento e gerenciamento das informações do jogo. Este projeto foi elaborado com o intuito de aplicar conhecimentos teóricos e práticos adquiridos ao longo do curso, abrangendo áreas como lógica de programação, orientação a objetos, manipulação de banco de dados e desenvolvimento de interfaces gráficas.

Durante o processo de desenvolvimento, foi necessário projetar toda a estrutura do jogo, incluindo o sistema de perguntas e respostas, o controle de pontuação, a interação com o usuário e a persistência dos dados em um banco relacional. A integração entre o front-end e o back-end foi realizada com o objetivo de garantir uma experiência fluida e funcional para os usuários.

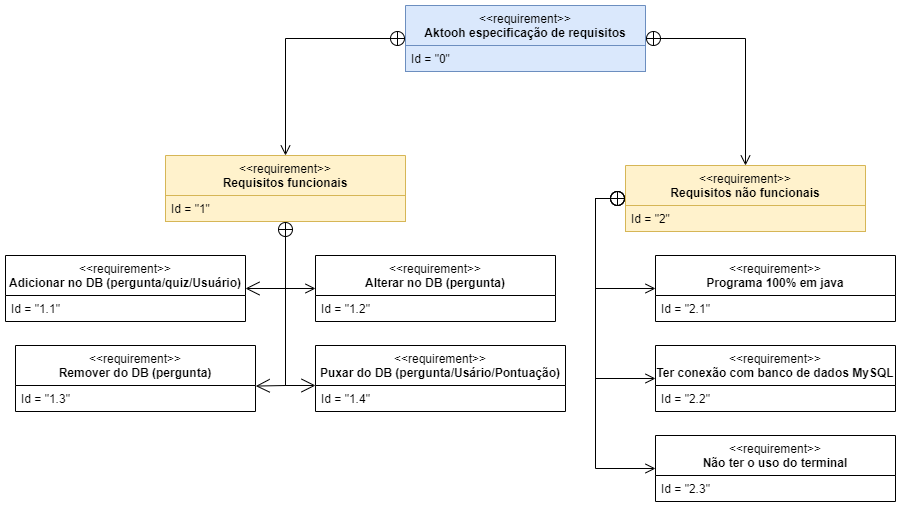
## 2. FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento de sistemas interativos baseados em quizes tem ganhado destaque no ambiente educacional por promover uma forma dinâmica e participativa de avaliação de conhecimento. O aplicativo Kahoot por exemplo se popularizou justamente por aliar a competitividade e o aprendizado em uma interface acessível e atrativa. Essas plataformas operam a partir de estruturas de perguntas e respostas com pontuações em tempo real, o que exige uma arquitetura de software capaz de garantir agilidade, armazenamento seguro e interatividade.

A linguagem de programação Java foi a linguagem proposta a ser utilizada neste projeto. Por ser uma linguagem orientada a objetos, o Java permite uma estruturação clara e modular do código, facilitando a organização das funcionalidades do sistema em uma possível manutenção futura. Esse paradigma favorece a reutilização de componentes do código, o encapsulamento de dados e a separação de responsabilidades, o que contribui para um desenvolvimento mais eficiente e organizado. A interface gráfica do sistema foi desenvolvida utilizando os recursos básicos disponíveis na própria linguagem Java, como a biblioteca Swing, que, embora mais simples em comparação a outras soluções gráficas modernas, atendeu às necessidades do projeto possibilitando a criação de telas interativas e funcionais.

A integração entre Java e o MySQL foi realizada por meio da API JDBC (Java Database Connectivity), que permite a conexão da aplicação com o banco de dados, a execução das Querys e o tratamento de resultados dentro do próprio código Java. Essa conexão é fundamental para garantir que as informações do sistema como usuários, quizzes, perguntas e resultados sejam armazenados e recuperados de forma eficiente e segura.

**Figura 1** – Requisitos do sistema



**Fonte:** Autores.

A comunicação entre o front-end (interface gráfica e lógica do jogo) e o back-end (responsável pelo armazenamento e recuperação de dados) é essencial para assegurar o funcionamento fluido do sistema, permitindo que as informações sejam acessadas e atualizadas de forma consistente e em tempo real.

## 3. METODOLOGIA

O desenvolvimento do projeto foi realizado em grupo por conta disso foi utilizado o sistema de controle de versões Git, por meio da plataforma GitHub, o que permitiu que cada integrante pudesse contribuir simultaneamente com alterações e melhorias nas classes do jogo, assim mantendo a organização e integridade do projeto.

A linguagem de programação Java foi adotada por exigência da proposta do projeto. A escolha do banco de dados relacional MySQL também seguiu essa obrigatoriedade, sendo utilizado para armazenar informações relevantes como os dados dos usuários, perguntas, respostas e pontuações. A comunicação entre a aplicação e o banco de dados foi implementada por meio da biblioteca JDBC, cuja configuração pode ser observada na classe ConnFactory, responsável pela criação e gerenciamento da conexão com o banco de dados.

Para a codificação, foi utilizada a IDE Visual Studio Code (VS Code). As classes e interfaces gráficas foram desenvolvidas com base na biblioteca Swing, que foi empregada para a criação das telas de login, realização do quiz, edição de perguntas e visualização dos resultados, como evidenciado nos arquivos GuiCentralAluno.java, GuiSetQuiz.java, GuiEditQuestions.java, entre outros.

A metodologia de desenvolvimento seguiu um modelo incremental, com testes constantes ao final de cada etapa para garantir o bom funcionamento do sistema. Inicialmente, foi feita a modelagem das funcionalidades principais, seguida da implementação da estrutura do banco de dados e, posteriormente, da criação das interfaces gráficas e da lógica de integração entre os módulos. A equipe se organizou de forma que cada membro ficasse responsável por diferentes aspectos do projeto, como a lógica do jogo, persistência dos dados, navegação entre telas e controle de usuários.

## 4. DESENVOLVIMENTO DO SISTEMA

Diagrama

O conteúdo gerado por IA pode estar incorreto.**Figura 2 –** Diagrama do Sistema

**Fonte:** Autores

O sistema desenvolvido tem como objetivo principal permitir que alunos participem de quizes interativos e que professores possam gerenciar os conteúdos aplicados, de forma prática e organizada. Para isso, o sistema foi dividido em diferentes módulos, cada um responsável por uma funcionalidade específica, interligados por meio de classes Java e integrados a um banco de dados relacional em MySQL.

### 4.1 Estrutura Geral do Sistema

A aplicação foi estruturada em torno de interfaces gráficas desenvolvidas com a biblioteca Swing, por meio das quais o usuário pode interagir com o sistema. Foram criadas sete GUI’s (Graphical User Interface), que são um tipo de interface que permite aos usuários interagirem com um computador através de elementos gráficos, como ícones, botões, janelas e menus, em vez de interfaces de linha de comando baseadas em texto. As principais interfaces incluem:

* GuiUser: tela inicial para login e cadastro de usuários.
* GuiCentralAluno.java: tela principal destinada ao aluno, com opções para iniciar quizes.
* GuiCentralProfessor: interface voltada ao professor, com opções de gerenciamento.
* GuiSetQuiz: utilizada para configurar e iniciar um quiz.
* GuiEditQuestions: permite ao professor cadastrar e editar perguntas.
* GuiViewResults e GuiViewResultsProfessor: exibem os resultados do quiz ao aluno e ao professor, respectivamente.
* GuiQuestions: exibe as perguntas do quiz, contendo 4 alternativas uma com uma cor e forma respectiva.

Outras classes criadas que não envolvem a interface gráfica do jogo são:

* Question: Representa as perguntas do quiz, com atributos para ID, texto da pergunta, opções de resposta e índice da resposta correta. Inclui métodos para verificação de respostas (isCorrect()).
* QuestionForm: Janela de diálogo para criação/edição de perguntas, com campos para texto da pergunta, 4 opções de resposta e seleção da resposta correta via JComboBox.
* ShapeIcon: Implementação personalizada de ícones para os botões de resposta (triângulo, diamante, círculo e quadrado), seguindo o estilo visual do Kahoot.
* User: Modelo de dados para usuários, com ID, nome, senha e pontuação acumulada. Possui construtores para novos usuários (sem ID) e usuários existentes.
* CrudBD: atua como a camada de acesso aos dados do sistema, atuando como intermediário entre a aplicação Java e o banco de dados MySQL, centralizando todas as operações de persistência, incluindo o cadastro e autenticação de usuários, o gerenciamento completo do ciclo de vida das perguntas (criação, leitura, atualização e exclusão), a organização de quizes personalizados e o armazenamento dos resultados obtidos pelos alunos.
* QuizApp: É responsável por inicializar o sistema e exibir a tela inicial de login (GuiUser). A partir dela, todo o fluxo da aplicação é desencadeado conforme o tipo de usuário autenticado (aluno ou professor). Sua função principal é garantir o carregamento da interface e a fluidez da navegação entre as telas, servindo como base para o funcionamento geral do sistema.

### 4.2 Funcionalidades Implementadas

O sistema conta com um conjunto de funcionalidades que abrangem desde o gerenciamento de usuários até a execução completa de um quiz, como descrito a seguir:

* **Cadastro e login**: Os usuários são identificados por meio de login e senha. As informações são armazenadas e validadas no banco de dados MySQL.
* **Gerenciamento de perguntas**: Professores podem criar, editar e excluir perguntas e alternativas. Essas ações são realizadas por meio da interface GuiEditQuestions.java.
* **Execução do quiz**: Alunos acessam as perguntas do quiz em GuiSetQuiz.java, respondem às questões e recebem a pontuação ao final.
* **Visualização de resultados**: Alunos e professores podem visualizar os resultados obtidos em quizes já concluídos.
* **Formulário de Perguntas**: A classe QuestionForm permite:
  + Preenchimento de texto e opções via campos de texto (JTextField).
  + Seleção da resposta correta através de um JComboBox.
  + Validação de campos obrigatórios antes do salvamento.
* **Integração com Banco de Dados**: As perguntas criadas/alteradas são persistidas via CrudBD.addQuestion() e CrudBD.updateQuestion(). Sendo que CrudBD é responsável por **realizar todas as operações de acesso ao banco de dados** do sistema, como inserir, consultar, atualizar e deletar dados. O nome "CRUD" vem justamente dessas quatro operações básicas:
  + **C** – *Create* (Criar): inserir novos dados no banco
  + **R** – *Read* (Ler): consultar dados
  + **U** – *Update* (Atualizar): modificar dados existentes
  + **D** – *Delete* (Excluir): remover dados

A sigla BD significa **Banco de Dados**. Logo, CrudBD é entendida como "Operações CRUD no Banco de Dados".

* **Ícones Personalizados**: A classe ShapeIcon enriquece a interface gráfica, atribuindo formas geométricas distintas a cada opção de resposta (triângulo para opção 1, diamante para opção 2, etc.).
* **Fluxo de Autenticação**: A classe QuizApp gerencia:
  + Cadastro de novos usuários (com ID aleatório gerado por CrudBD.saveUser()).
  + Login de usuários existentes com validação de senha.
  + Redirecionamento para interfaces específicas (aluno/professor).

### 4.3 Integração com Banco de Dados

O sistema utiliza o MySQL como banco de dados relacional para armazenar todas as informações necessárias ao funcionamento do quiz. A integração entre a aplicação Java e o banco de dados foi implementada através da API JDBC (Java Database Connectivity), seguindo um padrão de design que separa claramente as responsabilidades de acesso a dados da lógica de negócios.

A classe ConnFactory centraliza a configuração de conexão, fornecendo um método estático getConnection() que estabelece a ligação com o banco de dados "quizdb" usando credenciais pré-definidas (usuário "root" e senha "XXXXXX"). Esta abordagem segue o padrão Factory, garantindo que todas as partes do sistema utilizem a mesma configuração de conexão.

A estrutura do banco de dados foi projetada com cinco tabelas principais, cada uma com um propósito específico:

**users**: Armazena informações dos usuários (user\_id, name, senha, totalScore)

|  |  |  |
| --- | --- | --- |
| user\_id | name | senha |
| 11236 | Ruth | 12345 |
| 21156 | F | 12 |
| 29110 | Fe | 12 |
| 90310 | Felipe | 12345 |

**questions**: Contém todas as perguntas disponíveis (id, question, optionA, optionB, optionC, optionD, correctOption). Exemplo de uma parte da tabela a seguir:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| question | optionA | optionB | optionC | optionD | correctOption |
| Qual é a capital da França? | Paris | Londres | Roma | Berlim | 0 |
| Qual é o maior planeta do sistema solar? | Terra | Marte | Júpiter | Saturno | 2 |
| Quem escreveu "Dom Quixote"? | Machado de Assis | Miguel de Cervantes | José de Alencar | Eça de Queirós | 1 |
| Qual é o elemento químico representado pelo símbolo "O"? | Ouro | Oxigênio | Ozônio | Osmium | 1 |
| Quem pintou a Mona Lisa? | Vincent van Gogh | Michelangelo | Leonardo da Vinci | Pablo Picasso | 2 |
| Em que ano o homem pisou na Lua pela primeira vez? | 1965 | 1969 | 1972 | 1980 | 1 |
| Qual é o animal terrestre mais rápido do mundo? | Leopardo | Guepardo | Tigre | Cavalo | 1 |
| Qual é o menor país do mundo? | Mônaco | Malta | Vaticano | Liechtenstein | 2 |
| Qual linguagem é usada para desenvolver aplicativos Android nativos? | Java | Python | Swift | Kotlin | 0 |
| Quem desenvolveu a Teoria da Relatividade? | Isaac Newton | Nikola Tesla | Albert Einstein | Stephen Hawking | 2 |

**quizzes**: Registra os quizes criados (id, name)

|  |  |
| --- | --- |
| id | name |
| 1 | teste |
| 2 | teste |

**quiz\_questions**: Estabelece a relação muitos-para-muitos entre quizes e perguntas (quiz\_id, question\_id)

|  |  |
| --- | --- |
| quiz\_id | question\_id |
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 1 | 4 |
| 1 | 5 |
| 2 | 1 |
| 2 | 2 |
| 2 | 3 |
| 2 | 4 |
| 2 | 5 |
| 2 | 6 |
| 2 | 7 |

**results**: Armazena os resultados dos alunos (id, user\_id, quiz\_name, totalScore)

|  |  |  |  |
| --- | --- | --- | --- |
| id | user\_id | quiz\_name | totalScore |
| 1 | 90310 | Quiz Configurado | 890 |
| 2 | 90310 | Quiz Configurado | 945 |
| 5 | 21156 | Quiz Configurado | 966 |
| 6 | 29110 | Quiz Configurado | 938 |
| 7 | 29110 | Quiz Configurado | 928 |
| 9 | 90310 | Quiz Configurado | 934 |

A classe CrudBD encapsula todas as operações de banco de dados, utilizando PreparedStatements para prevenir injeção SQL e melhorar a segurança. As principais operações implementadas incluem:

* **Inserções (INSERT)**: Para criar registros de usuários, perguntas, quizes e resultados
* **Consultas (SELECT)**: Para recuperar listas de perguntas, quizes e resultados
* **Atualizações (UPDATE)**: Para modificar perguntas existentes e pontuações de usuários
* **Exclusões (DELETE)**: Para remover perguntas do sistema

Um exemplo importante é o método saveUser, que gera um ID aleatório entre 1 e 100 para novos usuários e utiliza a cláusula ON DUPLICATE KEY UPDATE para evitar duplicatas.

Figura 3 – Diagrama funcionamento do sistema

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Para o gerenciamento de perguntas, o sistema implementa operações CRUD completas. O método getRandomQuestions utiliza a função RAND() do MySQL para selecionar perguntas aleatórias, com limite definido pelo parâmetro.

A integração com o banco de dados também permite a criação de quizes personalizados, onde o método saveQuiz primeiro insere o quiz na tabela quizes e depois as associações com perguntas na tabela quiz\_questions.

Para garantir a consistência dos dados, todas as operações são realizadas dentro de blocos try-with-resources, que automaticamente fecham conexões, statements e result sets, prevenindo vazamentos de recursos. O tratamento de exceções SQLException é implementado em cada método, registrando erros no console para diagnóstico.

## 5. RESULTADOS E DISCUÇÃO

O desenvolvimento do sistema de quiz interativo, inspirado na plataforma Kahoot, resultou em um software funcional, que atendeu aos requisitos propostos inicialmente. O sistema permite que professores cadastrem perguntas, organizem quizzes personalizados e acompanhem o desempenho dos alunos, enquanto os alunos podem acessar o sistema, participar dos quizes e visualizar seus resultados.

A utilização da biblioteca Swing para a criação da interface gráfica, apesar de suas limitações em comparação com bibliotecas mais modernas, mostrou-se adequada para os objetivos do projeto, permitindo a criação de telas intuitivas e funcionais. A integração com o banco de dados MySQL, também se mostrou eficiente.

Abaixo seguem-se figuras demonstrando o funcionamento do quiz:

Figura 4 – tela inicial de cadastro ao rodar o jogo

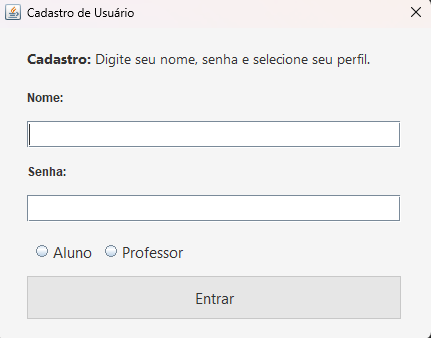


Figura 5 – Menu do jogo após cadastro

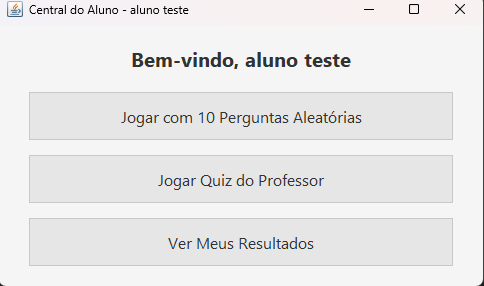
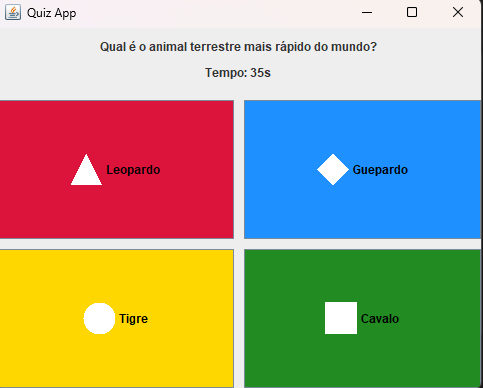


Figura 6 – exemplo de pergunta do quiz



## 6. CONSIDERAÇÕES FINAIS