

# Documentação de requisitos

## Índice:

- Introdução/setor.
- Config.Banco de dados/execução do cod.
- Requisitos funcionais
- Requisitos não funcionais
- Correção de código
- Papéis

- ***Introdução/setor.***

O objetivo dessa aplicação é facilitar o gerenciamento e o registro das notas dos alunos em um ambiente educacional. Ela permite que os professores insiram, visualizem e, se aplicável, editem as notas dos alunos de forma organizada e eficiente. Além disso, proporciona aos professores a capacidade de acessar as notas dos alunos utilizando seus nomes como referência. Essa aplicação irá melhorar a eficiência na administração de notas, substituindo métodos manuais e propensos a erros por um sistema automatizado e seguro.

**Setor:** educacional.

- ***Configuração do banco de dados, e execução do código.***

## Configuração do Banco de Dados:

Antes de executar o código, certifique-se de ter um banco de dados MySQL criado com o nome "seu\_banco\_de\_dados", e ajuste as variáveis "usuário" e "senha" no código com as credenciais corretas.

## Execução do Código:

Abra o projeto no eclipse.

Certifique-se de ter o conector JDBC para MySQL adicionado ao seu projeto.

Execute a classe "SistemaNotas" para iniciar o sistema.

- **Requisitos funcionais.**

**Cadastro de Professores e Alunos:** Funcionalidades implementadas. Os usuários podem cadastrar professores e alunos.

**Registro e Consulta de Notas:** As opções 2, 3 e 4 do menu abordam esses requisitos, permitindo o registro, visualização e atualização de notas.

**Edição de Notas:** Implementado na opção 4 do menu, permitindo que os professores atualizem as notas registradas.

- **Requisitos não funcionais.**

**Compatibilidade:** O código é executado em qualquer ambiente Java, garantindo compatibilidade multiplataforma.

**Segurança de Dados:** Garantia de segurança dos dados do aluno, incluindo a implementação de práticas de criptografia e proteção contra acessos não autorizados.

**Backup e Recuperação de Dados:** Implementação de rotinas de backup regulares e facilidade de recuperação em caso de perda de dados.

**Desempenho:** O sistema deve ser eficiente, garantindo tempos de resposta rápidos mesmo com um grande volume de dados.

**Interface:** A interface do sistema deve ser intuitiva e de fácil navegação para facilitar a inserção e consulta de notas.

**Documentação:** Elaboração de documentação detalhada do sistema para referência futura.

**Testes:** Desenvolvimento e execução de testes abrangentes para garantir a robustez e confiabilidade do sistema.

- **Correção de códigos.**

Visto que o grupo B identificou um erro no caso de tratamento de erros, conseguimos implementar um ajuste para deixar o código mais robusto para lidar melhor com entradas e saídas de dados inválidas. Onde já se encontra no git hub.

- **Papéis**

**Product owner:** Matheus Barbosa

1262223635

**Scrum master:** Felipe Neves 1262225380

**Equipe de dev:** Paulo Henrique 1262221832 Thiago pralon.1262217078