**Desenvolvimento Projeto Pessoal**

**Nome do Projeto:  
G.L.D.C. Gerenciamento Acadêmico**

**Nome do Criador:**

**Gabriel Lippi da Costa**

Sumário

[1. Descrição/Resumo do Projeto 4](#_Toc200271701)

[2. Requisitos Funcionais 4](#_Toc200271702)

[3. Requisitos Não-Funcionais 5](#_Toc200271703)

[4. Diagrama de Casos de Uso 5](#_Toc200271704)

[4.1. Diagrama de Classes 6](#_Toc200271705)

[4.3. Modelo de Banco de Dados 7](#_Toc200271706)

[4.4. Diagrama de Atividades 9](#_Toc200271707)

[5. Implementação 9](#_Toc200271709)

[6. Testes 9](#_Toc200271710)

[7. Referências 9](#_Toc200271711)

[8. Apêndice I 10](#_Toc200271712)

## Descrição/Resumo do Projeto

* O objetivo deste sistema é facilitar, organizar e otimizar o gerenciamento de cursos, alunos e professores em Instituições de Ensino.
* O sistema é voltado para funcionários autorizados da instituição, com permissões para realizar o controle e administração dessas informações.

Tecnologias utilizadas:

* **VSCode e NetBeans** – Desenvolvimento e edição de código-fonte.
* **MySQL** – Banco de dados.
* **Canva** – Criação de elementos gráficos e visuais.
* **Microsoft Word** – Documentação.

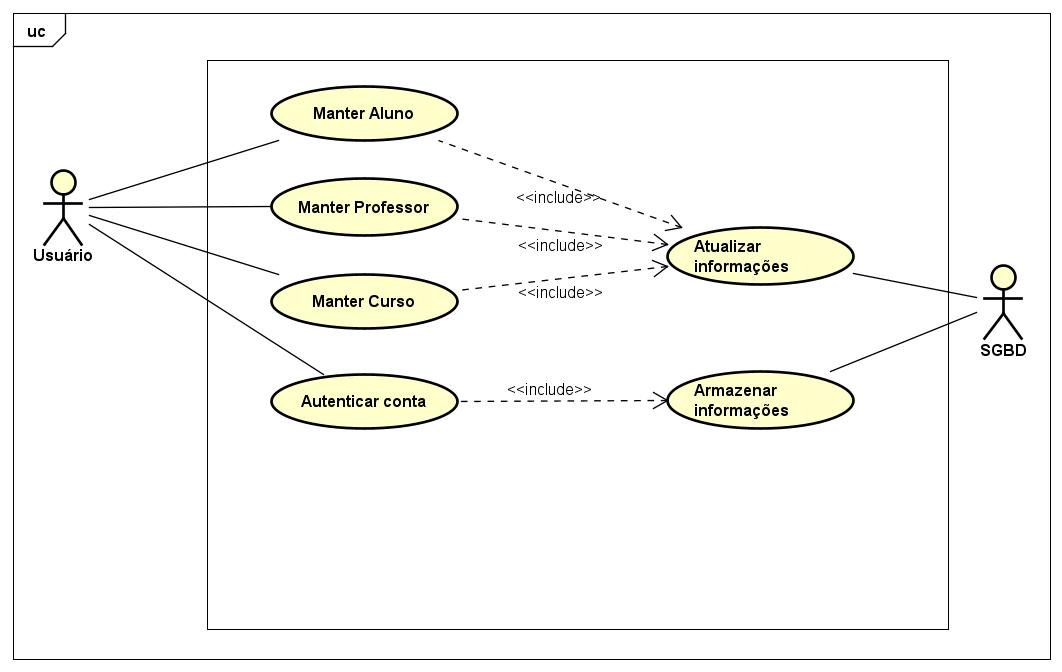
## 2. Requisitos Funcionais

|  |  |  |
| --- | --- | --- |
| ID | Descrição do Requisito | Prioridade |
| RF01 | O sistema deve permitir o gerenciamento de alunos. | Alta |
| RF02 | O sistema deve permitir o gerenciamento de professores. | Alta |
| RF03 | O sistema deve permitir o gerenciamento de cursos. | Alta |
| RF04 | O sistema deve armazenar informações sensíveis no banco de dados. | Alta |
| RF05 | O sistema deve funcionar apenas com conexão à internet. | Média |
| RF06 | O sistema deve validar a autenticação do usuário por e-mail e senha. | Alta |
| RF07 | O sistema deve exibir as informações de cursos, alunos e professores em formato de tabela. | Alta |

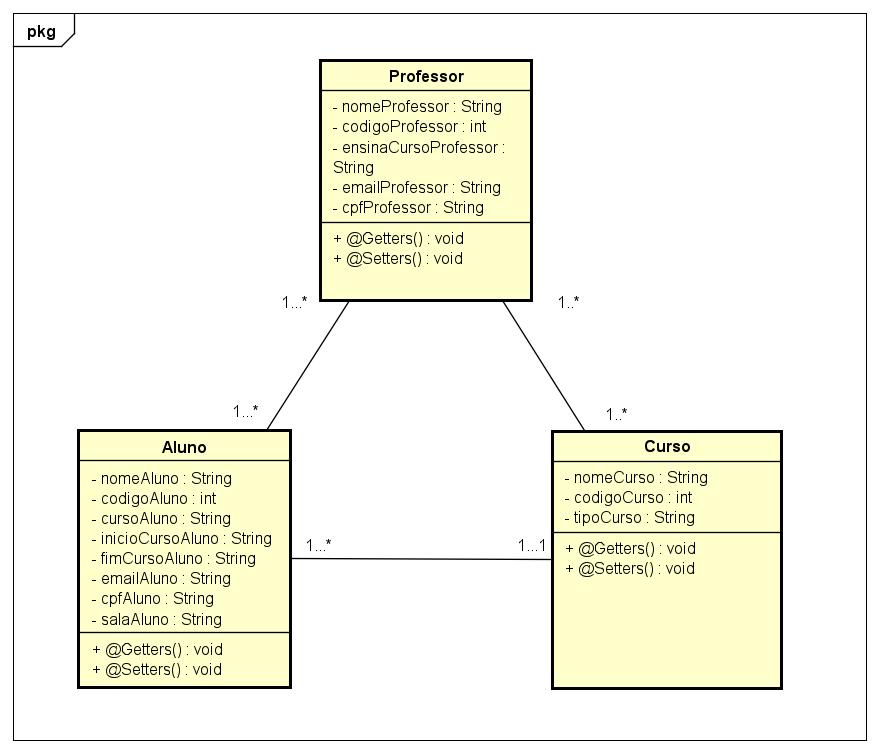
## 3. Requisitos Não-Funcionais

|  |  |  |
| --- | --- | --- |
| ID | Descrição do Requisito | Prioridade |
| RNF01 | O sistema deve ser acessível apenas em dispositivos como notebooks, PCs e desktops. | Alta |
| RNF02 | O sistema deve suportar até 100 usuários simultâneos. | Média |
| RNF03 | O sistema deve apresentar tempo de resposta de, no máximo, 5 segundos. | Alta |

## 4. Diagrama de Casos de Uso



## 4.1. Diagrama de Classes



## 4.3. Modelo de Banco de Dados

Interface gráfica do usuário, Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Script Banco de Dados:**

CREATE DATABASE gerenciamento\_academico;

USE gerenciamento\_academico;

CREATE TABLE tb\_usuario (

codigoUsuario INT PRIMARY KEY AUTO\_INCREMENT,

nomeUsuario VARCHAR(200) NOT NULL,

cpfUsuario varchar(14) NOT NULL UNIQUE,

dataNascimentoUsuario DATE NOT NULL,

emailUsuario VARCHAR(200) NOT NULL UNIQUE,

senhaUsuario VARCHAR(255) NOT NULL

);

CREATE TABLE tb\_cursos (

codigoCurso INT PRIMARY KEY AUTO\_INCREMENT,

nomeCurso VARCHAR(200) NOT NULL,

tipoCurso VARCHAR(200) NOT NULL,

codigoUsuario INT NOT NULL,

FOREIGN KEY (codigoUsuario) REFERENCES tb\_usuario(codigoUsuario)

);

CREATE TABLE tb\_aluno (

codigoAluno INT PRIMARY KEY AUTO\_INCREMENT,

nomeAluno VARCHAR(200) NOT NULL,

cursoAluno INT NOT NULL,

inicioCursoAluno DATE NOT NULL,

fimCursoAluno DATE NOT NULL,

emailAluno VARCHAR(150) NOT NULL UNIQUE,

cpfAluno CHAR(11) NOT NULL UNIQUE,

salaAluno VARCHAR(50) NOT NULL,

codigoUsuario INT NOT NULL,

FOREIGN KEY (cursoAluno) REFERENCES tb\_cursos(codigoCurso),

FOREIGN KEY (codigoUsuario) REFERENCES tb\_usuario(codigoUsuario)

);

CREATE TABLE tb\_professor (

codigoProfessor INT PRIMARY KEY AUTO\_INCREMENT,

nomeProfessor VARCHAR(200) NOT NULL,

ensinaCursoProfessor INT NOT NULL,

emailProfessor VARCHAR(200) NOT NULL UNIQUE,

cpfProfessor CHAR(11) NOT NULL UNIQUE,

codigoUsuario INT NOT NULL,

FOREIGN KEY (ensinaCursoProfessor) REFERENCES tb\_cursos(codigoCurso),

FOREIGN KEY (codigoUsuario) REFERENCES tb\_usuario(codigoUsuario)

);

4.4. Diagrama de Atividades

## Diagrama O conteúdo gerado por IA pode estar incorreto.

## 5. Implementação

* Link para o repositório de código-fonte.
* Tecnologias utilizadas (linguagens, frameworks, padrões de arquitetura).

## 6. Testes

* Testes unitários, de integração e funcionais.
* Evidências como prints ou links para os testes.

## 7. Referências

Listar fontes utilizadas no formato ABNT.

## 8. Apêndice I

Materiais adicionais, como roteiros de entrevistas ou questionários.