

Pontifícia Universidade Católica de Campinas

Escola Politécnica

Engenharia de Software • Banco de Dados

Projeto NotaDez

Cadu Martinez Spadari - Felipe Nasser Coelho Moussa -
Laura Lugli Fonseca Pereira - Maria Júlia Souza de Oliveira
- Rafaela Jacobsen Braga

11 de novembro de 2025

1. Apresentação do Descritivo do Projeto

O projeto foi feito com o intuito de ajudar professores e educadores de forma prática e fácil a terem acesso a todas as ferramentas as quais necessitam para ajudarem nos afazeres educacionais do dia a dia, como notas, frequências e trabalhos entre outros.

Como principal objetivo, buscamos facilitar o trabalho das instituições de educação e dos professores, fazendo um site (web app) funcional e de fácil manuseio.

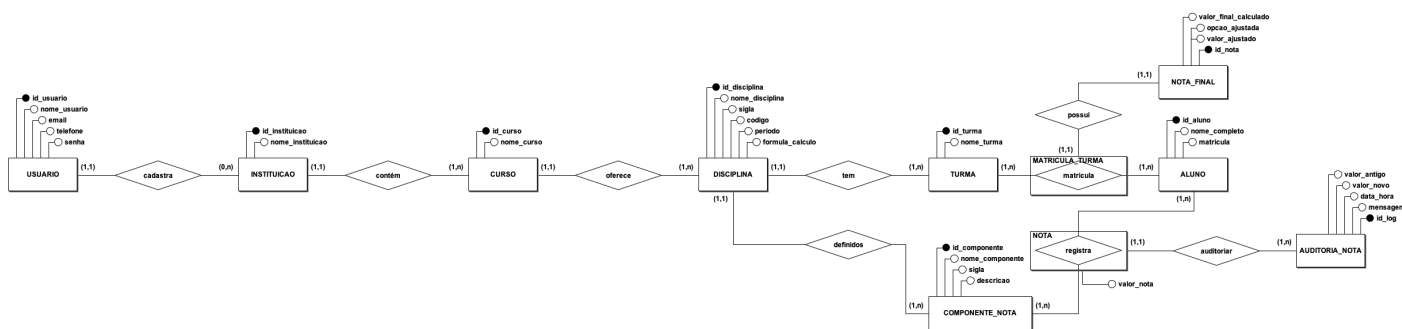
1.1. Liste os Requisitos Funcionais

O sistema permite que o usuário crie e acesse uma conta por meio de autenticação segura, com recuperação de senha e controle de acesso restrito a pessoas cadastradas. Após o login, o usuário pode gerenciar instituições, cursos, disciplinas e turmas, criando e excluindo cada elemento conforme a hierarquia estabelecida, com confirmações automáticas quando há dados associados. O sistema também possibilita o cadastro manual ou a importação de alunos a partir de arquivos CSV ou JSON, garantindo a integridade dos dados e evitando duplicidades.

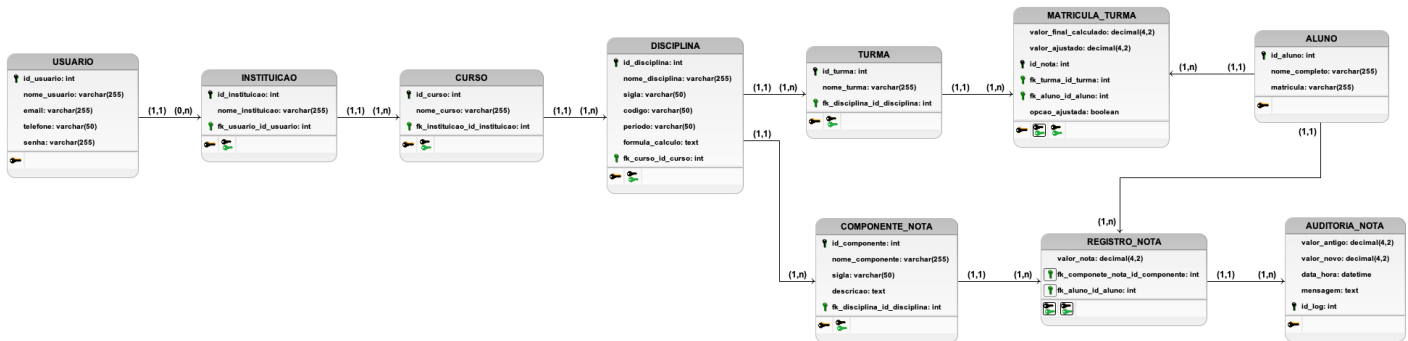
Além disso, o usuário pode definir e personalizar componentes de nota, configurando fórmulas próprias para o cálculo final de cada disciplina. As notas podem ser lançadas individualmente ou em modo de edição completa, sendo todas as alterações registradas automaticamente em um painel de auditoria que exibe data, hora e autor das modificações. O sistema realiza o cálculo automático das notas finais com base nas fórmulas definidas, e oferece a opção de adicionar uma coluna de “notas finais ajustadas”, permitindo arredondamentos automáticos e edições manuais dentro de critérios predefinidos.

Por fim, o sistema disponibiliza a exportação dos dados de notas em formato CSV ou JSON, garantindo que apenas informações completas e válidas sejam exportadas, assegurando consistência e confiabilidade nos resultados apresentados.

2. Modelagem do Sistema – Modelo Conceitual (MER)



3. Fazer o Modelo Lógico (usar softwares de modelagem)



4. Projeto de Banco de Dados – Modelo Físico (Criar no SQL developer – Colocar a estrutura da tabela criada).

```
CREATE TABLE USUARIO (
    id_usuario INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nome_usuario VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    telefone VARCHAR(50),
    senha VARCHAR(255) NOT NULL
);

CREATE TABLE INSTITUICAO (
    id_instituicao INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nome_instituicao VARCHAR(255) NOT NULL,
    fk_usuario_id_usuario INT UNSIGNED NOT NULL
);

CREATE TABLE CURSO (
    id_curso INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nome_curso VARCHAR(255) NOT NULL,
    fk_instituicao_id_instituicao INT UNSIGNED NOT NULL
);

CREATE TABLE DISCIPLINA (
    id_disciplina INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nome_disciplina VARCHAR(255) NOT NULL,
    sigla VARCHAR(50),
    codigo VARCHAR(50),
    periodo VARCHAR(50),
    formula_calculo TEXT,
    fk_curso_id_curso INT UNSIGNED NOT NULL
);

CREATE TABLE TURMA (
    id_turma INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nome_turma VARCHAR(255) NOT NULL,
    fk_disciplina_id_disciplina INT UNSIGNED NOT NULL
);

CREATE TABLE ALUNO (
    id_aluno INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nome_completo VARCHAR(255) NOT NULL,
    matricula VARCHAR(255) NOT NULL
);

CREATE TABLE MATRICULA_TURMA (
    id_matricula INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    id_turma INT UNSIGNED NOT NULL,
    id_aluno INT UNSIGNED NOT NULL,
    valor_final_calculado DECIMAL(4,2),
    valor_ajustado DECIMAL(4,2)
);

CREATE TABLE COMPONENTE_NOTA (
    id_componente INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nome_componente VARCHAR(255) NOT NULL,
    sigla VARCHAR(50),
    descricao TEXT,
    fk_disciplina_id_disciplina INT UNSIGNED NOT NULL
);

CREATE TABLE REGISTRO_NOTA (
    id_registro INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    id_componente INT UNSIGNED NOT NULL,
    id_aluno INT UNSIGNED NOT NULL,
    valor_nota DECIMAL(4,2)
);

CREATE TABLE AUDITORIA_NOTA (
    id_auditoria INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    id_registro INT UNSIGNED NOT NULL,
    valor_antigo DECIMAL(4,2),
    valor_novo DECIMAL(4,2),
    data_hora DATETIME,
    mensagem TEXT,
    id_log INT UNSIGNED NOT NULL
);
```

```

CREATE TABLE COMPONENTE_NOTA (
    id_componente INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nome_componente VARCHAR(255) NOT NULL,
    sigla VARCHAR(50),
    descricao TEXT,
    fk_disciplina_id_disciplina INT UNSIGNED NOT NULL
);

CREATE TABLE REGISTRO_NOTA (
    valor_nota DECIMAL(4,2),
    fk_componente_nota_id_componente INT UNSIGNED,
    fk_aluno_id_aluno INT UNSIGNED,
    fk_turma_id_turma INT UNSIGNED,
    PRIMARY KEY (fk_componente_nota_id_componente, fk_aluno_id_aluno)
);

CREATE TABLE AUDITORIA_NOTA (
    valor_antigo DECIMAL(4,2),
    valor_novo DECIMAL(4,2),
    data_hora DATETIME,
    mensagem TEXT,
    id_log INT UNSIGNED AUTO_INCREMENT PRIMARY KEY
);

CREATE TABLE MATRICULA_TURMA (
    valor_final_calculado DECIMAL(4,2),
    valor_ajustado DECIMAL(4,2),
    opcao_ajustada BOOLEAN,
    fk_turma_id_turma INT UNSIGNED,
    fk_aluno_id_aluno INT UNSIGNED,
    PRIMARY KEY (fk_turma_id_turma, fk_aluno_id_aluno)
);

ALTER TABLE INSTITUICAO ADD CONSTRAINT FK_INSTITUICAO_2
    FOREIGN KEY (fk_usuario_id_usuario)
    REFERENCES USUARIO (id_usuario)
    ON DELETE CASCADE;

ALTER TABLE CURSO ADD CONSTRAINT FK_CURSO_2
    FOREIGN KEY (fk_instituicao_id_instituicao)
    REFERENCES INSTITUICAO (id_instituicao)
    ON DELETE RESTRICT;

ALTER TABLE DISCIPLINA ADD CONSTRAINT FK_DISCIPLINA_2
    FOREIGN KEY (fk_curso_id_curso)
    REFERENCES CURSO (id_curso)
    ON DELETE RESTRICT;

ALTER TABLE TURMA ADD CONSTRAINT FK_TURMA_2
    FOREIGN KEY (fk_disciplina_id_disciplina)
    REFERENCES DISCIPLINA (id_disciplina)
    ON DELETE RESTRICT;

ALTER TABLE COMPONENTE_NOTA ADD CONSTRAINT FK_COMPONENTE_NOTA_2
    FOREIGN KEY (fk_disciplina_id_disciplina)
    REFERENCES DISCIPLINA (id_disciplina)
    ON DELETE RESTRICT;

```

```
ALTER TABLE REGISTRO_NOTA ADD CONSTRAINT FK_REGISTRO_NOTA_1  
    FOREIGN KEY (fk_componente_nota_id_componente)  
    REFERENCES COMPONENTE_NOTA (id_componente);
```

```
ALTER TABLE REGISTRO_NOTA ADD CONSTRAINT FK_REGISTRO_NOTA_2  
    FOREIGN KEY (fk_aluno_id_aluno)  
    REFERENCES ALUNO (id_aluno);
```

```
ALTER TABLE MATRICULA_TURMA ADD CONSTRAINT FK_MATRICULA_TURMA_2  
    FOREIGN KEY (fk_turma_id_turma)  
    REFERENCES TURMA (id_turma);
```

```
ALTER TABLE MATRICULA_TURMA ADD CONSTRAINT FK_MATRICULA_TURMA_3  
    FOREIGN KEY (fk_aluno_id_aluno)  
    REFERENCES ALUNO (id_aluno);
```

5. Considerações Finais

O trabalho foi desenvolvido de forma progressiva e organizada, com divisão de tarefas e uso de ferramentas colaborativas. As etapas de análise e modelagem foram cumpridas dentro do cronograma estipulado, resultando em um projeto com fundamentos sólidos, pronto para futuras integrações e testes completos. O comprometimento coletivo permitiu alcançar todos os objetivos propostos, demonstrando domínio das ferramentas utilizadas e uma execução exemplar de todo o processo de desenvolvimento.

Dito isso, houveram dificuldades sistêmicas quanto à criação das páginas html em si, além da modelagem de banco de dados, precisando ser revista e refeita com frequência, testada e reestruturada antes de chegar em uma versão final. Também houveram problemas de comunicação por parte do GitHub como condição de houver também dificuldade no manuseio da ferramenta.