

**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL - SENAC**

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADOR**

**Maestro - Conexão Musical**

**Porto Alegre**

**2025**



**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL - SENAC**

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADOR**

**MAESTRO - CONEXÃO MUSICAL**

**JOÃO PEDRO DA PAZ SAMPAIO DUTRA**

**Porto Alegre**

**2025**

**AGRADECIMENTOS**

XXXXXXXXXXXXXXXXX

.

**1 INTRODUÇÃO**

**Plataforma que auxilia professores e alunos para aprendizado virtual**

Maestro é um app feito para conectar quem quer aprender música com quem ensina. Se você é aluno e quer encontrar uma escola de música que combine com o seu estilo, o Maestro te ajuda a fazer isso sem complicação. Já para escolas e professores, é uma forma prática de organizar turmas, se comunicar com os alunos e divulgar seus cursos.

1. **Apresentação Geral do Projeto**

Maestro – Conexão Musical é um aplicativo desktop criado para aproximar pessoas que desejam aprender música de escolas especializadas. A plataforma permite que escolas se cadastrem informando sua localização e áreas de ensino, enquanto alunos criam perfis e são conectados a instituições próximas com base em seus interesses e região.

O processo de seleção da escola, pode ser feito diretamente no aplicativo. Após ingressar, o aluno terá acesso a um ambiente virtual organizado, onde poderá acompanhar conteúdos, atividades e seu progresso de aprendizado musical.

O Maestro – Conexão Musical promove um novo jeito de estudar música: acessível, interativo e conectado à realidade dos alunos.

1. **Objetivos**

**2.1 Objetivo Geral**

Desenvolver uma plataforma digital que oferece um ambiente virtual de apoio ao ensino musical, além de conectar escolas de música a estudantes interessados, facilitando o processo de descoberta e contato.

**2.2 Objetivos Específicos**

Criar um sistema de cadastro e localização para escolas de música, permitindo que elas insiram dados como especialidades, horários e região de atendimento.

Desenvolver um mecanismo de correspondência entre alunos e escolas próximas, com base na geolocalização e nas preferências musicais dos usuários.

Oferecer um ambiente de aprendizagem virtual estruturado, com funcionalidades que permitam aos alunos acompanhar aulas, materiais didáticos e tarefas de maneira organizada.

1. **JUSTIFICATIVA**

Há 8 anos estou morando em Porto Alegre e recentemente percebi que existem poucas escolas de música nos bairros da cidade. Além disso, somente as encontrei com um processo de busca na internet específico e difícil. Com meu projeto quero suavizar a busca para dar visibilidade a essas instituições de ensino e também facilitar para os alunos, para que eles não precisem enfrentar as mesmas dificuldades de encontrar uma escola que atenda aos seus desejos. Supondo que essas instituições sofrem com a falta de ferramentas digitais, meu projeto oferece um acesso a uma educação musical de qualidade através do espaço virtual na aplicação que é compreensível e acessível para o aluno e para a escola.

1. **Metodologia**

Durante o desenvolvimento do sistema será usado uma metodologia ágil, o que significa que o projeto será feito por partes. Isso ajuda a testar as funcionalidades aos poucos, corrigir erros rapidamente e adaptar o sistema conforme as necessidades forem aparecendo. Assim, o projeto vai evoluindo de forma mais leve e flexível, com melhorias constantes no uso por alunos, professores e escolas.

1. **Aquitetura do Sistema (Modelagem)**

**5.1 Requisitos**

**5.1.1 Requisitos Funcionais**

RF01 O sistema deve permitir o cadastro de escolas de música com nome da instituição, email, senha e localização.

RF02 O sistema deve permitir o cadastro de alunos com nome, email e senha.

RF03 O sistema deve possibilitar que escolas visualizem os alunos matriculados próximos de sua localização.

RF04 O sistema deve permitir que os alunos vejam as escolas disponíveis em sua região.

RF05 O sistema deve permitir que o aluno envie uma solicitação de matrícula para a escola escolhida.

RF06 O sistema deve permitir que as escolas aceitem ou recusem as solicitações de matrícula.

RF07 O sistema deve permitir login para alunos e escolas com verificação de email e senha.

RF08 O sistema deve exibir uma área de envios do professor com os conteúdos. organizados

RF09 O sistema deve permitir que escolas adicionem conteúdos, como vídeos, PDFs e links, aos envios..

**5.1.2 Requisitos Não Funcionais**

RNF01 O sistema deve possuir uma interface gráfica amigável e intuitiva,

RNF02 O banco de dados será implementado em SQLite, garantindo leveza e portabilidade do sistema.

RNF03 As senhas devem ser armazenadas de forma segura, evitando exposição de dados sensíveis.

RNF04 O design do app tem que ser viável, com janelas bem organizadas e acessíveis.

RNF05 O sistema deve estar preparado para futura expansão de funcionalidades, como chat interno ou videochamadas.

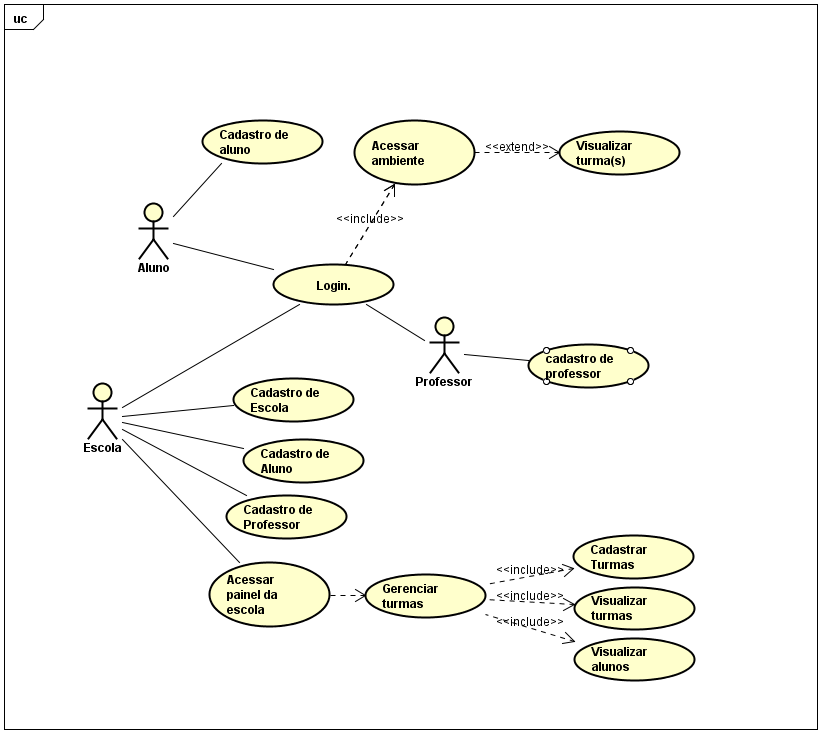
RNF06 O uso do sistema deve ser acessível para pessoas com diferentes níveis de conhecimento em informática..

**5.3 Modelagem Funcional**

**Lista dos Atores**

| **Ator** | **Descrição** |
| --- | --- |
| **Aluno** | Usuário principal da plataforma. Se cadastra para buscar e selecionar escolas próximas, realiza matrículas, acessa conteúdos e interage com professores no ambiente escolar virtual |
| **Professor** | Responsável por ministrar aulas e interagir com os alunos nas turmas vinculadas. Pode visualizar os alunos, acessar o ambiente virtual e futuramente publicar conteúdos. |
| **Escola** | Instituição de ensino musical que se cadastra na plataforma, informa sua localização, oferta cursos, aceita matrículas e administra seus professores e turmas. |

**5.4 Diagrama Casos de Uso**



**5.4.1 Detalhamento casos de uso**

Nesta seção será apresentado o detalhamento do seguintes casos de uso:

**CSU01 –** Login no sistema

**CSU02 –** Cadastro de escola

**CSU03 –** Cadastro de professor

**CSU04 –** Cadastro de aluno

**CSU05 –** Acesso ao painel da escola

**CSU06 –** Gerenciar turmas

**CSU07 –** Acesso ao ambiente do aluno

**CSU08 –** Visualizar turma

**CSU01 – Login no sistema**

| **Caso de Uso 1** | **Cadastro de Escola** |
| --- | --- |
| Objetivo | Permitir que alunos, professores ou representantes de escolas acessem o sistema com suas credenciais. |
| Ator | Aluno, Professor, Escola |
| Pré-condições | O usuário já deve estar cadastrado no sistema. |
| Cenário Principal  O usuário acessa a tela de login, insere email e senha. O sistema valida as credenciais e redireciona para o ambiente correspondente ao tipo de usuário (aluno, professor ou escola). | |
| Fluxo Alternativo  Caso os dados estejam incorretos, o sistema exibe uma mensagem de erro. O usuário pode tentar novamente. | |

**CSU02 – Cadastro de Escola**

| **Caso de Uso 2** | **Cadastro de Escola** |
| --- | --- |
| Objetivo | Permitir que uma nova escola musical se cadastre no sistema. |
| Ator | Escola |
| Pré-condições | Nenhuma. A escola ainda não está cadastrada. |
| Cenário Principal  A escola preenche nome, email, senha, endereço, telefone e outras informações. O sistema armazena os dados e confirma o cadastro. | |
| Fluxo Alternativo  Se o email já estiver cadastrado, o sistema informa erro. A escola pode tentar outro email. | |

**CSU03–Cadastro de Professor**

| **Caso de Uso 3** | **Cadastro de Professor** |
| --- | --- |
| Objetivo | Permitir que um novo professor seja cadastrado na plataforma, vinculado a uma escola. |
| Ator | Escola, Professor |
| Pré-condições | A escola deve estar logada. |
| Cenário Principal  A escola acessa o painel de solicitações para professores, confirma o cadastro. O professor é adicionado à base de dados. | |
| Fluxo Alternativo  Se o email já estiver sendo usado, o sistema alerta a escola. | |

**CSU04 – Cadastro de Aluno**

| **Caso de Uso 4** | **Cadastro de Aluno** |
| --- | --- |
| Objetivo | Permitir o cadastro de novos alunos. |
| Ator | Escola, Aluno |
| Pré-condições | O usuário acessa a área de cadastro. |
| Cenário Principal  Os dados são preenchidos (nome, email, senha, endereço, telefone). O sistema salva e confirma o cadastro. | |
| Fluxo Alternativo  Se faltar algum campo ou o email já estiver em uso, o sistema exibe erro. | |

**CSU05 –Acesso ao painel da escola**

| **Caso de Uso 5** | **Acesso ao ambiente administrativo da escola** |
| --- | --- |
| Objetivo | Permitir que a escola visualize e gerencie seus dados no painel. |
| Ator | Escola |
| Pré-condições | A escola deve estar logada. |
| Cenário Principal  Após login, a escola acessa o painel com abas para visualizar professores, alunos e turmas | |
| Fluxo Alternativo  Erro de conexão ou falha ao carregar dados são informados com mensagem. | |

**CSU06 – Gerenciar turmas**

| **Caso de Uso 6** | **Cadastrar Aeronave** |
| --- | --- |
| Objetivo | Permitir que a escola visualize turmas vinculadas e seus detalhes. |
| Ator | Escola |
| Pré-condições | A escola deve estar logada. |
| Cenário Principal  Na aba “Turmas”, o sistema exibe as turmas da escola com nome, descrição, professor e especialidade. | |
| Fluxo Alternativo  Se não houver turmas, o sistema exibe uma mensagem indicando ausência de dados. | |

**CSU07 –Acesso ao ambiente do aluno**

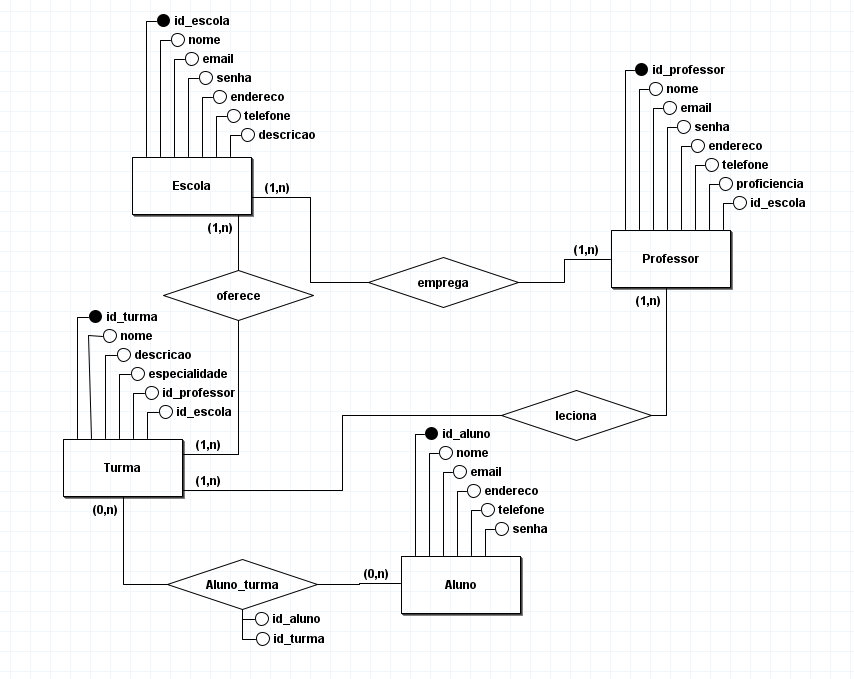
| **Caso de Uso 7** | **Acesso ao ambiente do aluno** |
| --- | --- |
| Objetivo | Permitir que o aluno visualize as turmas que participa. |
| Ator | Aluno |
| Pré-condições | O aluno deve estar logado. |
| Cenário Principal  O sistema exibe os cards das turmas com botão “Acessar”. Cada card mostra o nome da turma, descrição e especialidade. | |
| Fluxo Alternativo  Caso o aluno não esteja em nenhuma turma, o sistema exibe uma mensagem apropriada. | |

**CSU08 – Visualizar turma**

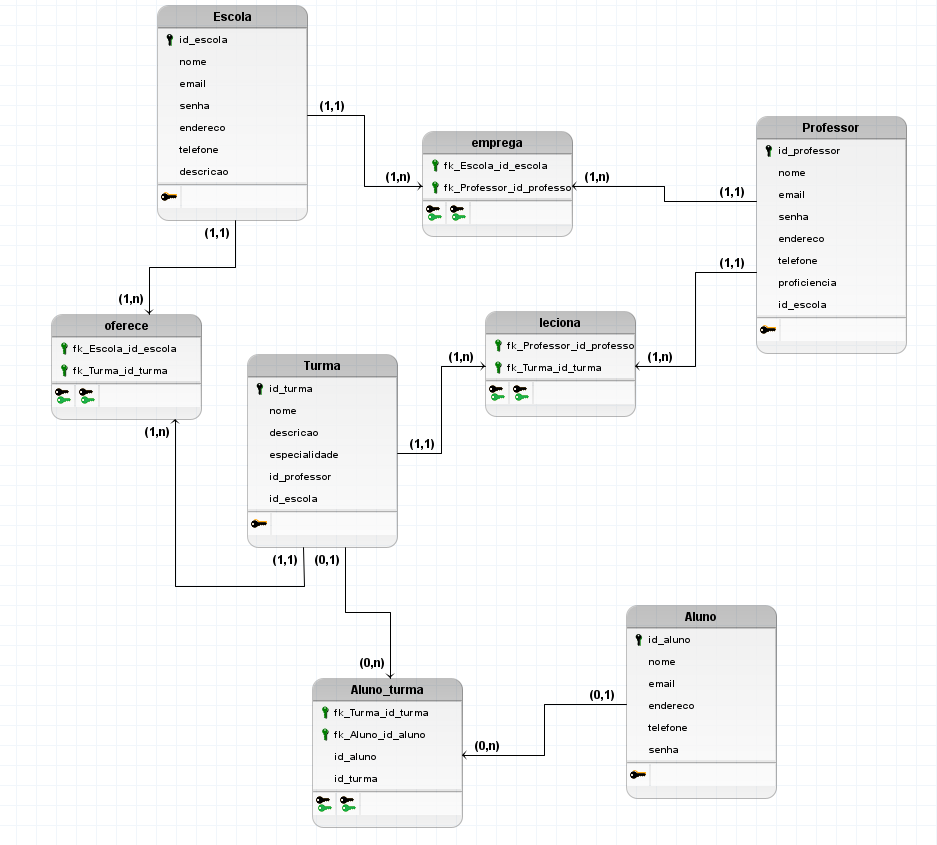
| **Caso de Uso 8** | **Visualizar turma** |
| --- | --- |
| Objetivo | Permitir que o aluno acesse uma turma específica e veja seu conteúdo. |
| Ator | Aluno |
| Pré-condições | O aluno precisa estar logado e inscrito na turma. |
| Cenário Principal  Ao clicar em “Acessar”, o sistema abre uma nova tela com abas “Mural” e “Pessoas”, mostrando a descrição da turma, professor e colegas. | |
| Fluxo Alternativo  Se a turma não existir mais ou houver erro de carregamento, o sistema informa o aluno. | |

1. **BANCO DE DADOS**

**6.1 Modelo Conceitual**



**6.2 Modelo Lógico**



**6.3 Modelo físico**

import sqlite3

conn = sqlite3.connect('escola.db')

cursor = conn.cursor()

cursor.execute("""

CREATE TABLE IF NOT EXISTS Escola (

id\_escola INTEGER PRIMARY KEY AUTOINCREMENT,

nome TEXT NOT NULL,

email TEXT NOT NULL UNIQUE,

senha TEXT NOT NULL,

endereco TEXT,

telefone TEXT,

descricao TEXT

)

""")

cursor.execute("""

CREATE TABLE IF NOT EXISTS Professor (

id\_professor INTEGER PRIMARY KEY AUTOINCREMENT,

nome TEXT NOT NULL,

email TEXT NOT NULL UNIQUE,

senha TEXT NOT NULL,

endereco TEXT,

telefone TEXT,

proficiencia TEXT,

id\_escola INTEGER NOT NULL,

FOREIGN KEY (id\_escola) REFERENCES Escola(id\_escola)

)

""")

cursor.execute("""

CREATE TABLE IF NOT EXISTS Turma (

id\_turma INTEGER PRIMARY KEY AUTOINCREMENT,

nome TEXT NOT NULL,

descricao TEXT,

especialidade TEXT,

id\_professor INTEGER NOT NULL,

id\_escola INTEGER NOT NULL,

FOREIGN KEY (id\_professor) REFERENCES Professor(id\_professor),

FOREIGN KEY (id\_escola) REFERENCES Escola(id\_escola)

)

""")

cursor.execute("""

CREATE TABLE IF NOT EXISTS Aluno (

id\_aluno INTEGER PRIMARY KEY AUTOINCREMENT,

nome TEXT NOT NULL,

email TEXT NOT NULL UNIQUE,

endereco TEXT,

telefone TEXT,

senha TEXT NOT NULL

)

""")

cursor.execute("""

CREATE TABLE IF NOT EXISTS Aluno\_Turma (

id\_aluno INTEGER NOT NULL,

id\_turma INTEGER NOT NULL,

PRIMARY KEY (id\_aluno, id\_turma),

FOREIGN KEY (id\_aluno) REFERENCES Aluno(id\_aluno),

FOREIGN KEY (id\_turma) REFERENCES Turma(id\_turma)

)

""")

conn.commit()

conn.close()

1. **PROJETO PYTHON**

**7.1 Descrição**

O projeto Maestro foi desenvolvido em Python, utilizando a biblioteca Tkinter para construção da interface gráfica e o SQLite como banco de dados local. A estrutura permite o cadastro e gerenciamento de usuários (escolas, professores e alunos), bem como o controle de turmas e acesso ao ambiente virtual de aprendizagem. A instalação do Python é necessária para executar o sistema, sendo recomendada a versão mais recente disponível no site oficial. Durante o processo de instalação, é importante marcar a opção “Add Python to PATH”.

**7.2 Projeto com CRUD**

**7.2.1 Telas**

A seguir estão as principais telas do sistema e suas respectivas funções no fluxo de uso.

* **Página Inicial**

Permite ao usuário acessar o sistema com suas credenciais ou realizar o cadastro de uma nova conta (Aluno, Professor ou Escola).

É a primeira tela exibida quando o programa é iniciado. Contém dois botões principais: “Login” e “Cadastro”, a opção cadastro tem três ramificações para um cadastro específico sendo eles “Aluno, “Professor”, “Escola”

* **Sessões de Cadastro** Tela onde o usuário escolhe se deseja cadastrar uma Escola, Professor ou Aluno. Após a escolha, é direcionado ao formulário específico. Facilita o cadastro de novos usuários no sistema, armazenando os dados no banco de dados.
* **Tela de Conexão com Escolas**

Essa tela pode ser acessada por alunos ainda não vinculados a uma escola. Nela é possível procurar instituições cadastradas em um mapa interativo, com base na localização. Exibe escolas próximas, com dados como nome, descrição, cidade e estado. Permite ao aluno escolher com qual escola deseja se conectar ou obter mais informações.

* **Tela de Contato com a Escola**

Após o aluno escolher uma escola, ele tem acesso a essa tela, onde são exibidas informações detalhadas de contato, como número de telefone, e-mail institucional e breve descrição da escola.  
 Permite ao aluno entrar em contato direto com a instituição, geralmente por ligação ou mensagem externa. Serve como passo final para efetivar a matrícula, que será feita pela própria escola fora do sistema.

* **Ambiente da Escola**

Ambiente de administração exibido para escolas logadas. Exibe três abas: Professores, Alunos e Turmas. Permite visualizar e gerenciar professores, alunos e turmas associadas à instituição.

* **Ambiente do Aluno**

Interface personalizada exibida para alunos logados. Exibe as turmas às quais o aluno está matriculado.  
 Permite que o aluno visualize informações das turmas, acesse murais e veja os participantes.

**7.2.2 Conexão com Banco**

O banco de dados utilizado é o SQLite, armazenado localmente no arquivo escola.db. A conexão é realizada com a biblioteca **sqlite3**, e os dados são manipulados por meio de comandos SQL executados diretamente no código. As tabelas são criadas automaticamente caso não existam

**7.3 Manual de utilização**

Ao abrir o sistema Maestro, o usuário visualiza a tela inicial com as opções de Login e Cadastro. Escolas, professores e alunos podem se cadastrar e, em seguida, realizar login com seu e-mail e senha.

Usuários com perfil de escola são direcionados ao ambiente administrativo, onde podem visualizar turmas, alunos e professores vinculados à instituição. Alunos, ao logar, visualizam uma interface parecida com a do Google Classroom, com cards representando as turmas nas quais estão matriculados. Clicando em uma turma, têm acesso ao mural e à lista de pessoas participantes.

Todos os dados são armazenados localmente no banco SQLite, permitindo que as informações sejam preservadas entre sessões. O sistema é leve, simples e de fácil utilização.

1. **PROJETO WEB**

**8.1 Telas HTML e CSS**

**8.2 Crud web**

1. **PLANO DE TESTE**

**8.1 Plano de Teste Desktop**

**8.2 Plano de Teste Web**

1. **CONCLUSÃO**
2. **REFERÊNCIAS**