

**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL - SENAC**

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADOR**

**ACCESS**

**Porto Alegre**

**2024**



**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL - SENAC**

**CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADOR**

**ACCESS**

**JULIO FRISKI**

**Porto Alegre**

**2024**

**AGRADECIMENTOS**

XXXXXXXXXXXXXXXXX

.

**1 INTRODUÇÃO**

**Sistema de Organização de Pequenos Eventos**

## 

## Apresentação Geral do Projeto

O *Access* tem como principal objetivo auxiliar os usuários na organização de pequenos eventos sociais, profissionais, corporativos, educacionais e até culturais.

A aplicação consiste em um simples sistema onde qualquer usuário pode criar o seu próprio evento, tornando-se assim, automaticamente o administrador desse evento, o administrador é quem tem a função de enviar os convites personalizados com os links para participar e entrar na lista. Os convidados que abrirem o link, ao entrar na página, irão se deparar com alguns requisitos impostos pelo criador, e assim, ao preenchido, podem solicitar a entrada no evento, que será aceita, ou não pelo administrador.

## Objetivos

### 2.1 Objetivo Geral

Auxiliar os usuários na organização de eventos mais simples, simplificando essa problemática.

### 2.2 Objetivos Específicos

Montar listas para eventos com base nas solicitações do usuário;

Criar convites personalizados;

Sistema de check-in com QRCode(Opcional);

Avaliar o evento e o criador;

## JUSTIFICATIVA

Percebi esse problema, o que me incentivou a criar essa aplicação

## Metodologia

Metodologia ágil, especificamente a Scrum, onde vou gerenciar o projeto em ciclos curtos (sprints), com foco em entregas rápidas, colaboração e adaptação constante

## Aquitetura do Sistema (Modelagem)

## 5.1 Requisitos

### 5.1.1 Requisitos Funcionais

RF01. O sistema deve realizar autenticação do usuário.

RF02. O sistema deve permitir ao usuário cadastrar eventos.

RF03. O sistema deve permitir ao usuário entrar em eventos.

RF04. O sistema deve permitir ao usuário consultar seus últimos eventos.

RF05. O sistema deve permitir ao usuário avaliar o evento.

RF06. O sistema deve permitir ao administrador deletar evento.

RF07. O sistema deve permitir ao administrador remover o participante.

RF08. O sistema deve permitir ao administrador gerenciar as solicitações(recusar ou aceitar).

RF09. O sistema deve permitir ao administrador compartilhar o link do convite de entrada para o evento.

RF10. O sistema deve fornecer um QR code exclusivo para cada convidado que entra na lista de um evento.

RF11. O sistema deve permitir ao administrador fazer check-in pelo QR code

### 5.1.2 Requisitos Não Funcionais

RNF01. A base de dados deve ser protegida para acesso apenas para usuários autorizados;

RNF02. O sistema deve ser desenvolvido utilizando a tecnologia de programação Kotlin para aparelhos Android e Swift para aparelhos IOS;

RNF03. A interface deve ser intuitiva e fácil de usar por pessoas sem conhecimento técnico;

RNF04. O sistema deve utilizar o banco de dados SQLite;

RNF05. O sistema deve estar disponível 99,9% do tempo durante o mês;

RNF06. O código-fonte deve seguir padrões para facilitar manutenção.

## 

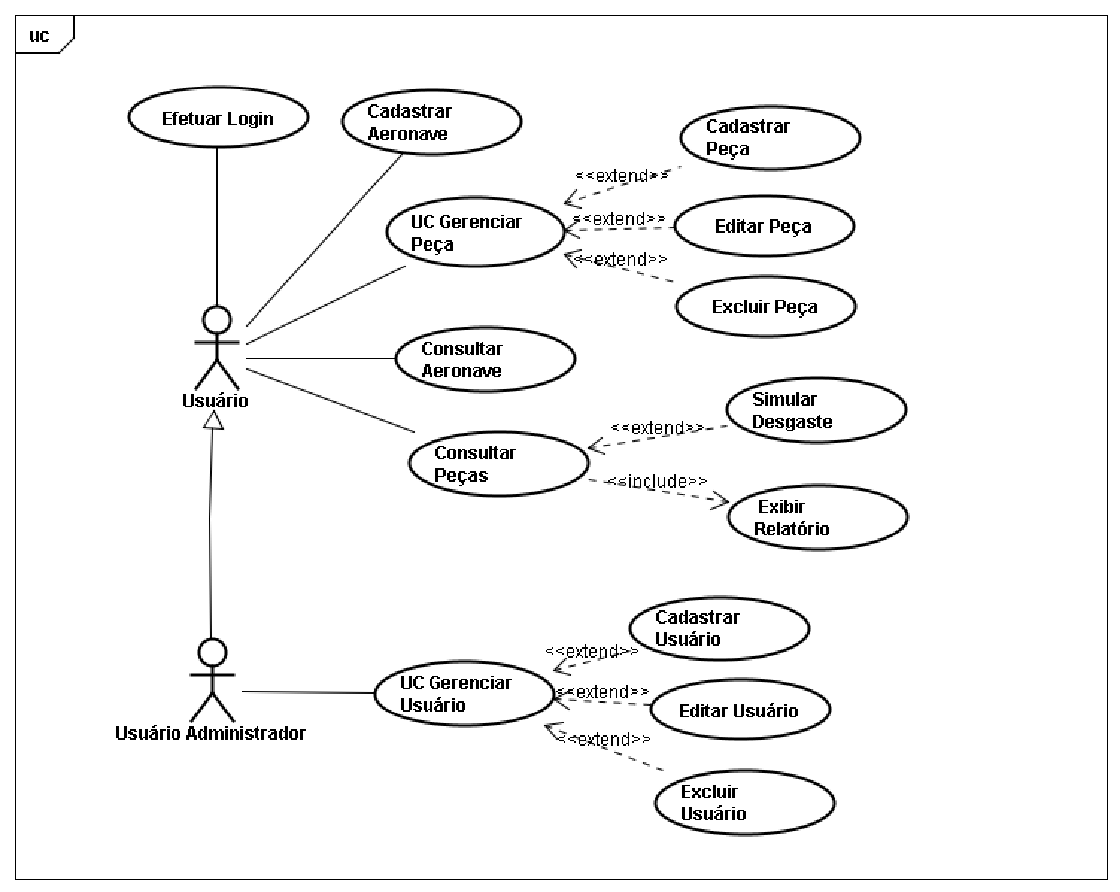
## 5.3 Modelagem Funcional

### Lista dos Atores

| **Ator** | **Descrição** |
| --- | --- |
| **Usuário Administrador** | O administrador do evento é responsável por gerenciar as funções superiores do evento.  É responsável por deletar evento, remover, aceitar ou recusar participantes |
| **Usuário** | O usuário é responsável por fazer login e por entrar no evento.  O usuário acessa o sistema para entrar em novos eventos, consultar os recentes, avaliar e também criar eventos. |

### 

### 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 – Efetuar Login

CSU02 – Cadastro

CSU03 – Criar Evento

CSU04 – Entrar em um Evento

CSU05 – Avaliar Evento

CSU06 – Consultar Histórico

CSU07 – Gerenciar Solicitações

CSU08 – Deletar Evento

CSU09 – Remover Participante

**CSU01 – Efetuar login**

| **Caso de Uso 1** | **Efetuar Login** |
| --- | --- |
| Objetivo | Tem por objetivo oferecer segurança de acesso à ferramenta, permitindo acesso apenas por usuários cadastrados. |
| Ator | Usuário |
| Pré-condições | Usuário estar cadastrado |
| Cenário Principal   1. O usuário entra com dados para login; 2. O sistema verifica informações do usuário; 3. O sistema inicia sessão relacionada ao usuário; | |

**CSU02 – Cadastro**

| **Caso de Uso 2** | **Cadastro** |
| --- | --- |
| Objetivo | Tem por objetivo cadastrar-se no aplicativo |
| Ator | Usuário |
| Pré-condições | Deve ser um novo usuário sem conta |
| Cenário Principal   1. Novo usuário entra no aplicativo e inicia o cadastro; 2. O usuário preenche campos para cadastro; 3. Sistema armazena usuário; 4. Sistema retorna mensagem de sucesso para cadastro de usuário. | |
| Fluxo Alternativo   1. Algum dado é inválido: 2. Um aviso é exibido e proíbe efetuar o cadastro. 3. Usuário clica no botão de voltar em meio ao cadastro: 4. Sistema retorna à tela de login. | |

**CSU03– Criar Evento**

| **Caso de Uso 3** | **Criar Evento** |
| --- | --- |
| Objetivo | Tem por objetivo criar um novo evento |
| Ator | Usuário |
| Pré-condições | O usuário deve estar logado |
| Cenário Principal   1. Administrador clica no botão de “Novo Evento”; 2. Administrador preenche campos requisitados; 3. Sistema armazena o evento; 4. Sistema retorna mensagem de sucesso para cadastrar evento. | |
| Fluxo Alternativo   1. Usuário cancela operação em qualquer um dos passos 2. Sistema retorna à tela inicial. | |

**CSU04 – Entrar em um Evento**

| **Caso de Uso 4** | **Entrar em um Evento** |
| --- | --- |
| Objetivo | Tem por objetivo entrar em um evento |
| Ator | Usuário |
| Pré-condições | O usuário tem que estar logado |
| Cenário Principal   1. Usuário entra em um convite recebido; 2. É redirecionado para a tela de entrar em evento; 3. Clica no botão “Entrar no Evento” 4. Sistema retorna mensagem de sucesso e o usuário vai para a lista de espera. | |
| Fluxo Alternativo   1. Usuário cancela operação em qualquer um dos passos 2. Sistema retorna para a tela inicial 3. Convite inválido.   1. Retorna uma mensagem de convite inválido e vai para a tela inicial | |

**CSU05 – Avaliar Evento**

| **Caso de Uso 5** | **Avaliar Evento** |
| --- | --- |
| Objetivo | Tem por objetivo avaliar um evento; |
| Ator | Usuário |
| Pré-condições | Estar logado;  Ser um participante do evento;  O evento já ter ocorrido. |
| Cenário Principal  1. Usuário entra no histórico de eventos;  2. Usuário avalia de 1 a 5 estrelas o evento e o organizador;  3. Sistema cadastra a avaliação e avisa o usuário por meio de uma mensagem; | |
| Fluxo Alternativo   1. Usuário cancela operação em qualquer um dos passos 2. Sistema retorna para a tela inicial   B. Convite inválido.  1. Retorna uma mensagem de convite inválido e vai para a tela inicial | |

**CSU06 – Consultar Histórico**

| **Caso de Uso 6** | **Consultar Histórico** |
| --- | --- |
| Objetivo | Tem por objetivo consultar histórico de eventos participados |
| Ator | Usuário |
| Pré-condições | Estar logado;  Ser um participante do evento;  O evento já ter ocorrido. |
| Cenário Principal   1. Usuário seleciona o botão do histórico de eventos; 2. Sistema mostra os eventos que o usuário participou; | |
| Fluxo Alternativo  A. Usuário cancela operação em qualquer um dos passos   1. O sistema retorna à tela inicial.   B. Nenhum evento aparece pois o usuário não tem nenhum evento no histórico   1. Mensagem “Sem eventos” aparece | |

**CSU07 – Gerenciar Solicitações**

| **Caso de Uso 7** | **Gerenciar Solicitações** |
| --- | --- |
| Objetivo | Tem por objetivo aceitar ou recusar usuários em um evento; |
| Ator | Usuário Administrador do Evento |
| Pré-condições | Ser o criador do evento;  O usuário a ser analisado deve ter solicitado a entrada no evento. |
| Cenário Principal  1. Administrador entra no evento criado;  2. Administrador seleciona o botão de listar solicitações;  3. Sistema abre uma janela com o nome do usuário e alguma informação adicional;  4. Administrador clica em Aceitar ou Recusar o participante, se aceito, vai para a lista do evento, se recusado, vai para a lista de recusados do evento. | |
| Fluxo Alternativo  A. Administrador cancela operação em qualquer um dos passos   1. Sistema retorna para a tela do evento.   B. Não existe solicitações para entrar no evento   1. Mensagem de “Sem solicitações” é exibida ao abrir a janela de solicitações | |

**CSU08 – Deletar Evento**

| **Caso de Uso 8** | **Deletar Evento** |
| --- | --- |
| Objetivo | Tem por objetivo deletar um evento |
| Ator | Usuário Administrador do Evento |
| Pré-condições | Ser o criador do evento;  O evento deve ser gratuito;  Se pago, não pode ter nenhum convidado na lista e nenhuma solicitação com pagamento; |
| Cenário Principal   1. Administrador entra na tela do evento; 2. Seleciona o botão de deletar evento; 3. Sistema deleta o evento; 4. Sistema retorna mensagem de sucesso. | |
| Fluxo Alternativo   1. Usuário cancela operação em qualquer um dos passos 2. Retorna para a tela do evento 3. O evento não possui os requisitos necessários para ser deletado 4. Mensagem proibindo a remoção é exibida. | |

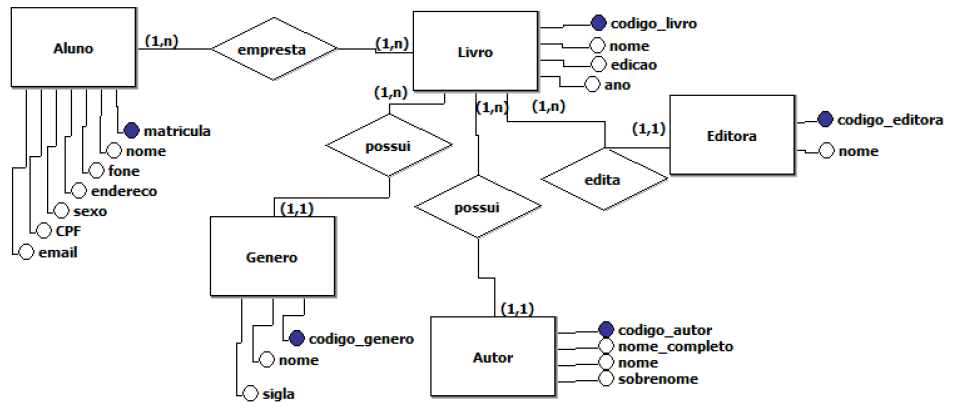
**CSU09 – Remover Participante**

| **Caso de Uso 9** | **Remover Participante** |
| --- | --- |
| Objetivo | Tem por objetivo Remover Participante |
| Ator | Usuário Administrador do Evento |
| Pré-condições | Ser o criador do evento;  O usuário a ser removido deve estar na lista do evento;  O evento não pode ser pago. |
| Cenário Principal   1. Administrador seleciona o usuário a ser removido; 2. Administrador preenche os dados da remoção; 3. Sistema atualiza as informações; 4. Sistema retorna mensagem de sucesso para remover usuário do evento. | |
| Fluxo Alternativo   1. Usuário cancela operação em qualquer um dos passos 2. O evento não atende os requisitos para ser possível remover o usuário. | |

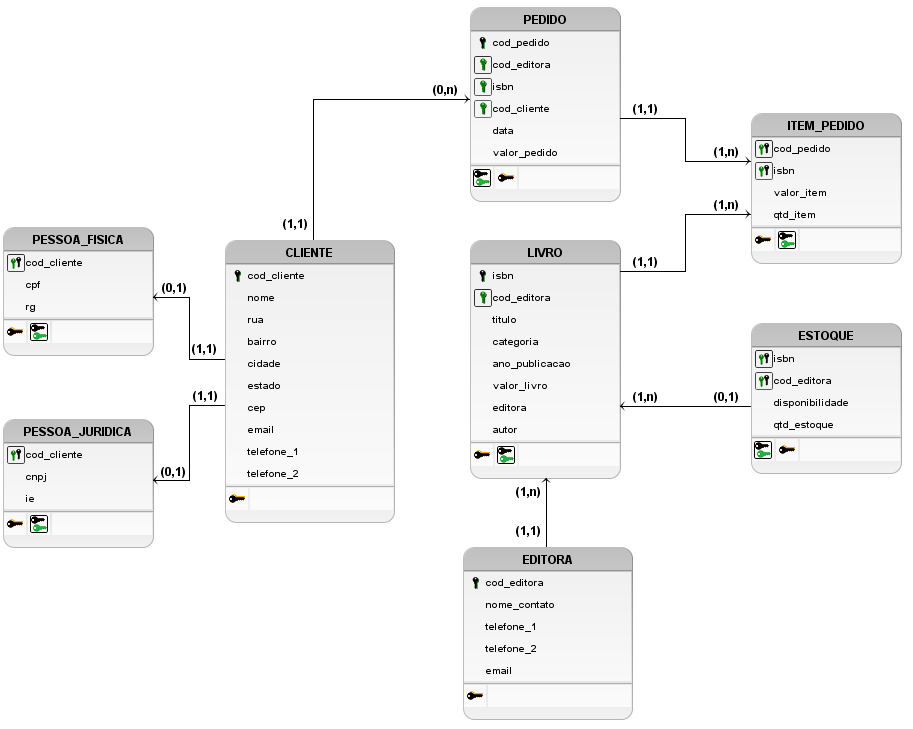
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  | |
|  | |

1. **BANCO DE DADOS**

### 6.1 Modelo Conceitual



### 6.2 Modelo Lógico



### 6.3 Modelo Físico

create database biblioteca;

use biblioteca;

create table editora(

ideditora int primary key auto\_increment not null,

nomeeditora varchar(30) not null

);

create table genero(

idgenero int primary key auto\_increment not null,

nomegenero varchar(30),

sigla char(3)

);

create table autor(

idautor int primary key auto\_increment not null,

nomecompleto varchar(60) not null,

nomeautor varchar(30) not null,

sobrenome varchar(30) not null

);

create table livro(

idlivro int primary key auto\_increment not null,

nomelivro varchar(50) not null,

edicao varchar(4) not null,

ano varhcar(4),

genero int not null,

autor int not null,

editora int null,

FOREIGN KEY (genero) REFERENCES genero (idgenero),

foreign key (autor) references autor (idautor),

foreign key (editora) references editora (ideditora)

);

create table aluno(

matricula varchar(30) primary key not null,

nomealuno varchar(50) not null,

fone varchar(11),

endereco varchar(100) not null,

genero varchar(15) not null,

cpf varchar(11) not null unique,

email varchar(100) not null

);

create table empresta(

matricula varchar(30) not null,

idlivro int not null,

foreign key (matricula) references aluno (matricula),

foreign key (idlivro) references livro (idlivro),

primary key (matricula, idlivro)

);

1. **PROJETO JAVA**

### 7.1 Tutorial de instalação do Java

### 7.2 Projeto com CRUD

### 7.2.1 Telas

### 7.2.2 Conexão com Banco

### 7.3 Manual de utilização

1. **PROJETO WEB**

### 8.1 Telasd 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**