

Documentação

Desenvolvido por:

Enzzo Lima

Erika Leite

Gabriel Pereira

Joyce Santos

Sumário

1.	Resumo.....	3
2.	Escopo do projeto	3
2.1.	Sistema Web	3
2.1.1.	Perfis de Usuário	3
2.1.2.	Funcionalidades	3
3.	Modelagem de software	4
3.1.	Modelo Conceitual	4
3.2.	Modelo Lógico	4
3.3.	Modelo Físico	5
3.4.	Cronograma.....	5
4.	Back-End.....	6
4.1.	Como executar e testar a API?	6
5.	UI / UX	7
5.1.1.	Imagens do layout	7
6.	Front-end	8
6.1.1.	Imagens da aplicação na web	8

1. Resumo

Uma nova escola deseja criar um sistema de gerenciamento de seu patrimônio, para facilitar a gestão de suas salas e equipamentos.

2. Escopo do projeto

Disponível em < https://miro.com/app/board/o9J_l_OcB_Q=/>.

A aplicação possui os seguintes requisitos:

2.1. Sistema Web

2.1.1. Perfis de Usuário

1. Administrador: Para o colaborador da área administrativa da escola;
2. Comum: Para colaboradores de almoxarifado realizarem o cadastro de novos equipamentos e salas.

2.1.2. Funcionalidades

1. O administrador poderá cadastrar qualquer tipo de usuário (administrador ou comum);
2. O administrador poderá cadastrar novos usuários à plataforma e realizar a gestão deles;
3. O administrador poderá excluir salas e equipamentos;
4. O administrador poderá atualizar dados de equipamentos, salas e usuários;
5. O usuário comum poderá inserir novos equipamentos ao sistema;
6. O administrador poderá gerar relatórios de controle;

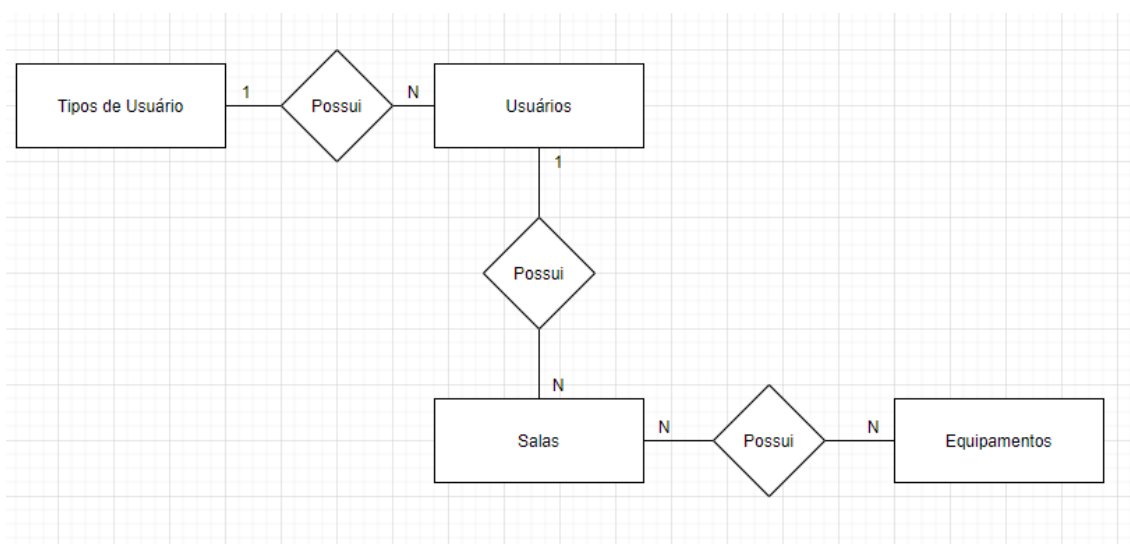
3. Modelagem de software

O banco de dados relacional é uma coleção de objetos, ou seja, são tabelas onde os dados são armazenados. Como exemplo, podemos querer armazenar dados sobre os produtos de um comércio. Para isso, criamos tabelas que guardam diferentes conjuntos de dados relacionados, como dados de fornecedores, dados de compras, dados de saída etc. Cada uma dessas tabelas é uma relação do banco.

De modo geral, a utilização de um banco de dados gera uma melhora na organização, facilita a gestão e o processamento dos dados. Isso acontece porque os dados são organizados de modo a modelar aspectos do mundo real, gerando informações relevantes para os usuários a partir desses dados.

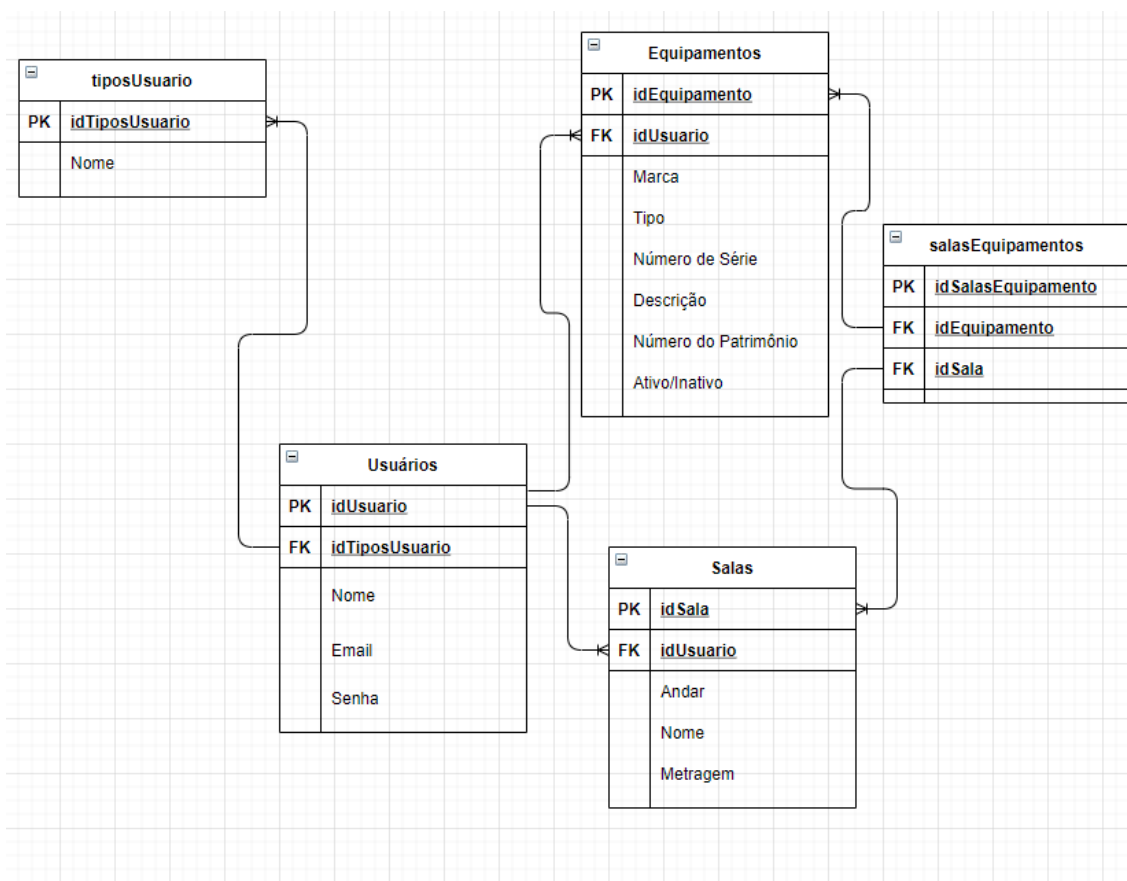
3.1. Modelo Conceitual

O modelo conceitual é um conjunto de suposições baseadas no mundo real que indicarão as regras de negócio de um sistema. Portanto, modelo conceitual é a descrição do sistema proposto na forma de um conjunto de ideias.



3.2. Modelo Lógico

Este modelo adequa o modelo conceitual para o tipo de banco de dados que será implementado. É desenvolvido na fase de projeto e não é de interesse dos usuários, somente da equipe de desenvolvimento. Nesse modelo que os atributos e as relações entre ficam mais evidentes.



3.3. Modelo Físico

Este modelo é o projeto físico para implementação do banco de dados. Tem todas as preocupações com desempenho, com a capacidade dos equipamentos etc. Pode-se utilizar o Excel para simular a estrutura real do banco de dados com base no modelo lógico elaborado anteriormente.

tiposUsuario		Usuários				
idTiposUsuario	Nome	idUsuario	Nome	Email	Senha	idTiposUsuario
1	Administrador	1	Administrador	adm@adm.com	ADM123	1
2	Comum	2	João	joao@hotmail.com	JOAO123	2
		3	Lucas	lucas@hotmail.com	LUCAS123	2

Salas					idUsuario
idSala	Andar	Nome	Metragem		
1	Primeiro	Sala1	25		2
2	Segundo	Sala2	40		2

salasEquipamentos		
idSalasEquipame	idSalas	idEquipamentos
1	1	3
2	1	4
3	2	2

Equipamentos						
idEquipamento	Marca	Tipo	Numero de Serie	Descricao	Numero do Patrimônio	Ativo/Inativo
1	Phillips	Eletroeletrônica	123456	Jogo de chaves	220456	1
2	Panasonic	Eletroeletrônica	785466	Câmera Digital	554023	0
3	HP	Informática	123879	Laptop	443522	1
4	Sony	Informática	456457	Fone de ouvido	456888	1

3.4. Cronograma

Todo o cronograma foi desenvolvido utilizando a metodologia Kanban e organizado pela equipe através do Trello.

Link do Trello: < <https://trello.com/b/GQBGFSQN/projeto-inicial-3-termo>>.

4. Back-End

O Back-End da aplicação, ou seja, o código do sistema, foi desenvolvido no Microsoft Visual Studio 2019 na linguagem C#. É uma API (Application Programming Interface) com arquitetura REST (Representational State Transfer), que é um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de padrões e protocolos. As requisições e respostas seguem o protocolo HTTP (HyperText Transfer Protocol) utilizando formato JSON (JavaScript Object Notation) para intercâmbio de dados, possibilitando o acesso do sistema em diversos dispositivos.

4.1. Como executar e testar a API?

São necessários os seguintes programas:

- Microsoft SQL Server;
- Microsoft SQL Management Studio;
- Microsoft Visual Studio;
- Postman.

5. UI / UX

O Layout da aplicação, tanto para web quanto para dispositivos móveis, foi desenvolvido no Figma. Para a criação do layout da aplicação foram feitos wireframes, tanto de baixa fidelidade quanto de alta fidelidade, e prototipação de todas as páginas, para saber se a interface e a experiência do usuário tivessem qualidade.

5.1.1. Imagens do layout

Web

The wireframes illustrate the user interface for the SENAI web application. The layout is organized into a grid with a central column and two side columns. The central column contains the main content areas, while the side columns contain navigation and auxiliary information.

Central Column Wireframes:

- Top Wireframe:** Displays a table of "Salas cadastradas" (Registered Rooms) with columns for Name, Address, and Action. Below it is a section for "Equipamentos cadastrados" (Registered Equipment).
- Middle Wireframe:** Contains the "Cadastrar equipamentos" (Register equipment) form, which includes fields for equipment type, brand, serial number, and description, along with an "Estado" (Status) dropdown and a "Cadastrar" button. Below this is the "Editar equipamentos" (Edit equipment) form with similar fields and an "Atualizar" button.

Side Column Wireframes:

- Left Side:** A "LOGIN" screen with fields for Email and Senha (Password), a "Esqueceu a senha?" link, and a "LOGIN" button.
- Right Side:** Contains three forms: "Cadastrar sala" (Register room), "Editar Sala" (Edit room), and "Cadastrar usuário" (Register user). Each form includes fields for Name, Address, and other relevant information, along with "Cadastrar" or "Editar" buttons.

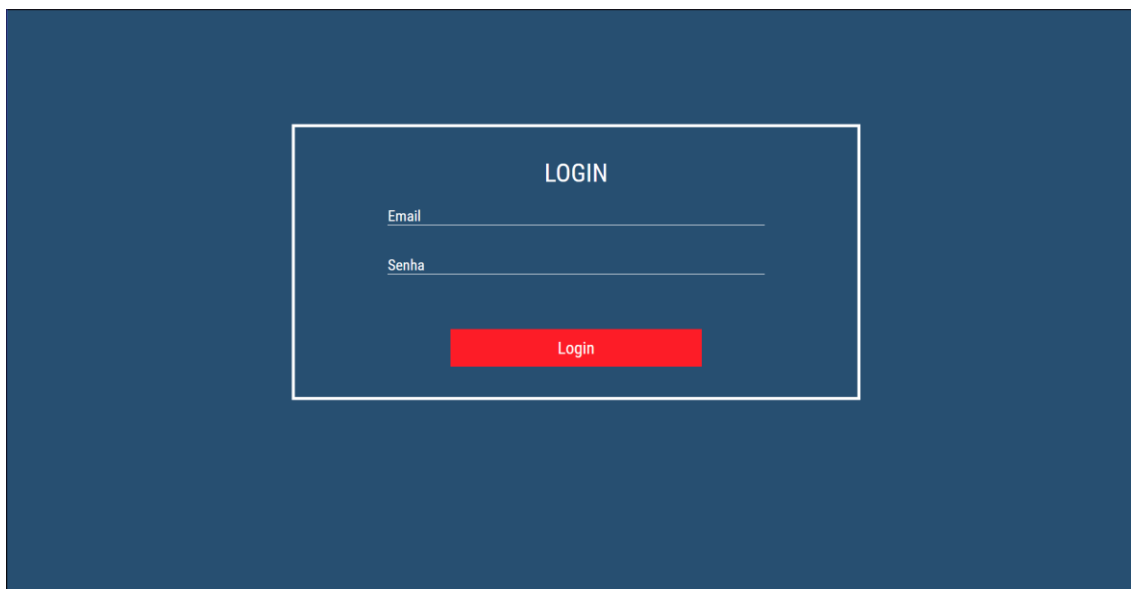
The wireframes use a clean, modern design with a color palette of red, white, and grey. The SENAI logo is prominently displayed at the top of each page.

6. Front-end

O Front-end da aplicação, ou seja, toda a interface utilizada pelo usuário final, foi desenvolvido no Microsoft Visual Studio Code 2019 utilizando a biblioteca React.

6.1.1. Imagens da aplicação na web

Página de Login



Home do sistema

Salas Cadastradas			
Nome	Andar	Ações	
Git	2ª	X	

Equipamentos Cadastrados			
Nº Patrimônio	Nome	Marca	Nº de Série
256348	Microondas	Philips	M34MK5D

Páginas de Cadastros e Edições

Cadastrar Sala

Nome Andar Metragem

Cadastrar

Editar Sala

Nome Andar Metragem

Editar

Cadastrar Equipamento

80% + Redefinir

Tipo de Equipamento Marca

Nº de Serie Nº de Patrimônio

Descrição

Estado

☐ Ativo ☐ Inativo

Cadastrar

Editar Equipamento

Tipo de Equipamento Marca

Nº de Serie Nº de Patrimônio

Descrição

Estado

☐ Ativo ☐ Inativo

Editar

Cadastrar Usuário

Tipo de Usuário Nome

Email Senha

Cadastrar

Editar Usuário

Tipo de Usuário Nome

Email Senha

Editar