Integrantes da Equipe

- Leandro de Aquino Rodrigues - 12724134626
- Emanuel Santos da Silva - 12724125621
- Matheus Azevedo de Oliveira - 12724129603
- Raissa Adorno Santos da Silva - 12724124545
- Pedro Ivo Barbosa Monzini - 1272411107
- Pedro Henrique Costa Damásio Matos - 12724121377

Descrição do Softwares necessários para execução da aplicação

Para executar a nossa aplicação, é necessário a instalação de dois softwares:

1. Visual Studio Code (VS Code) - Para editar e armazenar os códigos fontes da aplicação. Baixe e instale o VS Code para seu sistema operacional acessando https://code.visualstudio.com/.

2. NODE.JS - Permite que os desenvolvedores utilizem JS para criar aplicações web. Acesse https://nodejs.org/pt e baixe a versão recomendada (LTS), clique no botão "Descarregar a Node.js (LTS)" e execute o instalador (node-vxx.xx.x-x64.msi), depois clique em **Next** até o final e depois em **Install**.

Usamos diversas linguagens como HTML, CSS, JS e JSON, também usamos duas bibliotecas, a Axios e o React, além de uma ferramenta de código aberto, Prisma, que facilita a interação com o banco de dados em aplicações Node.js. Sobre o banco de dados, fizemos o seu código no MySQL e conectamos o banco online no RAILWAY, uma plataforma de hospedagem e desenvolvimento de aplicações em nuvem.

Instruções para instalação e execução da aplicação

1. Clonar ou baixar o projeto (back-end e front-end)

Faça o download das pastas com os códigos pelo link do GitHub: RaissaAdorno/sistema-reservas-restaurante

2. Abrir as pastas no VS Code

1. No VS Code, vá em **Arquivo > Abrir Pasta (File > Open Folder)**
   → Selecione a pasta backend

2. Vá em **Arquivo > Nova Janela (File > New Window)**
   → Em seguida, **Arquivo > Abrir Pasta**
   → Selecione a pasta frontend

3. Executar o back-end

1. Na janela do backend, vá em **Terminal > Novo Terminal (Ctrl + Shift + `)** e execute os comandos abaixo:

Passo 1 – Instalar dependências

npm install

Esse comando instala todas as dependências do projeto.

Erro comum: npm não reconhecido

Se aparecer a mensagem:

npm : O termo 'npm' não é reconhecido como nome de cmdlet...

Solução:

1. Feche o VS Code

2. No Windows, vá até a pasta onde o Node.js foi instalado (geralmente:
   C:\Program Files\nodejs)

3. Verifique se há o arquivo npm.cmd na pasta

4. Adicione essa pasta ao **PATH** do sistema:

   - Pressione **Win + R**, digite sysdm.cpl

   - Vá até a aba **Avançado** > clique em **Variáveis de Ambiente**

   - Em **Variáveis do sistema**, selecione Path e clique em **Editar**

     Adicione:

   - C:\Program Files\nodejs

5. Aplique e feche as janelas

Abra o **PowerShell como administrador** e execute:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

6. Confirme digitando **Y** (ou **S**)

Reabra o VS Code e tente novamente o comando:

```
npm install
```

Passo 2 – Gerar Prisma Client

npx prisma generate

Gera o cliente Prisma com base no schema do projeto.

Passo 3 – Iniciar o servidor

node --watch server.js

Inicia o servidor com "modo observação" (reinicia automaticamente ao detectar alterações nos arquivos).

4. Executar o front-end

1. Na janela do frontend, abra um terminal:
   Terminal > Novo Terminal (Ctrl + Shift + `)

2. Entre na pasta do projeto:

cd front-a3

3. Inicie o servidor de desenvolvimento:

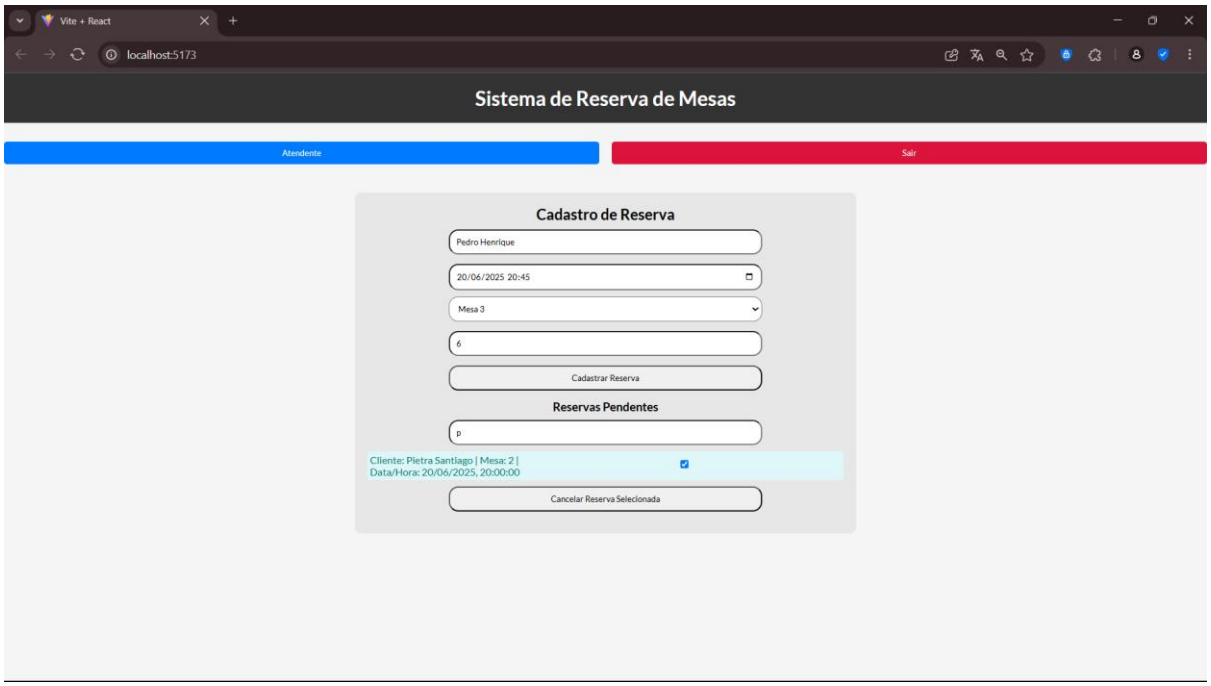
npm run dev

4. Após a execução, será exibido um link no terminal:
   Segure Ctrl e clique no link http://localhost:5173/ para abrir no navegador.

Com o site aberto, você verá tela inicial onde terá um sistema de login e senha, permitindo ao usuário realizar login como atendente, garçom ou gerente.

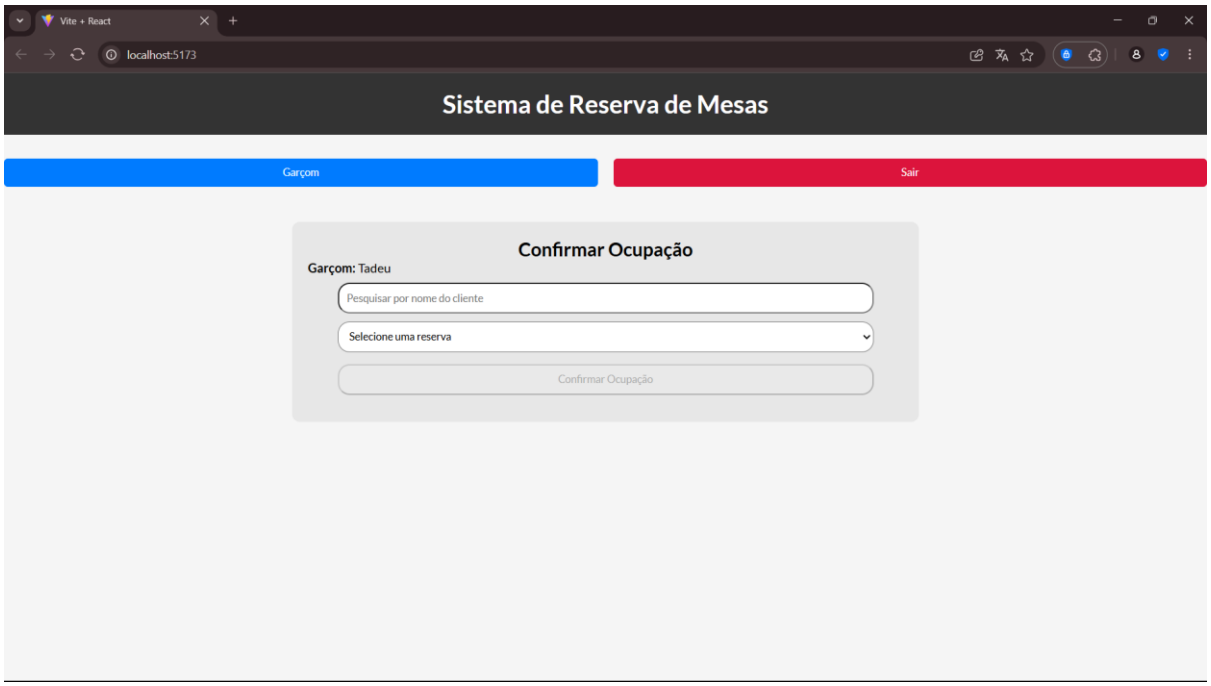
- Atendente – Login: Junior; Senha: 123
- Garçom – Login: Tadeu; Senha: 456
- Gerente – Login: Felca; Senha: 789

Caso o usuário entre como atendente, ele terá acesso a aba do atendente, onde ele poderá cadastrar as reservas dos clientes, colocando o nome do cliente, a data e a hora informadas, uma mesa a qual não esteja reservada por outro cliente e informando a quantidade de pessoas que estarão presentes. O atendente também poderá ver as reservas pendentes e cancelá-las se for necessário, pesquisando pelo nome do cliente.



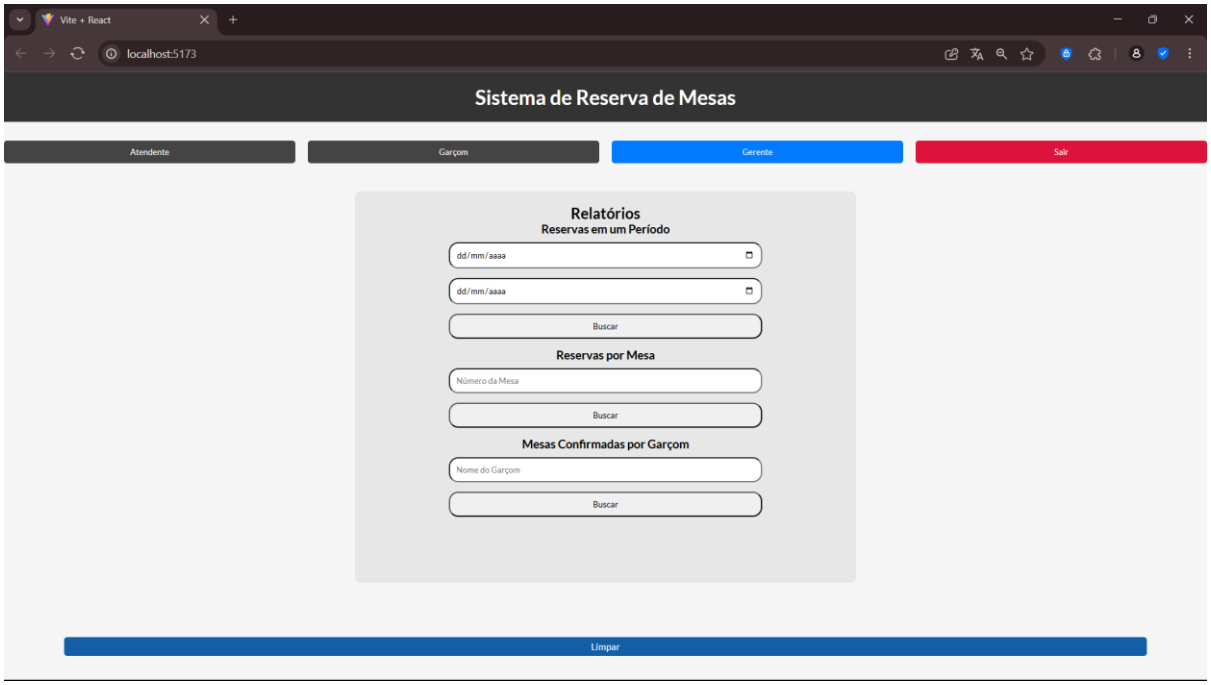
Caso entre como um garçom, o usuário terá acesso a aba do garçom, onde ele confirma as reservas cadastradas pelo atendente selecionando

a reserva desejada, podendo também pesquisar pelo nome ou primeira letra do cliente para filtrar.



Caso entre como gerente, o usuário terá acesso as abas do atendente e do garçom, podendo exercer suas funções. Além disso, ele terá acesso ao relatório do restaurante, podendo ver uma tabela detalhada com os dados das reservas solicitadas ao preencher pelo menos um desses campos:

- Reservas em um Período: São as reservas feitas em um determinado período do ano. (Ex: entre agosto e novembro do ano passado)
- Reservas por Mesa: Mostra as reservas feitas na mesa selecionada.
- Mesas Confirmadas por Garçom: Mostra as reservas confirmadas pelo garçom selecionado, é necessário digitar o nome dele. Lembre-se de que o gerente (Felca) pode exercer a função de garçom, ou seja, ele também pode confirmar reservas.



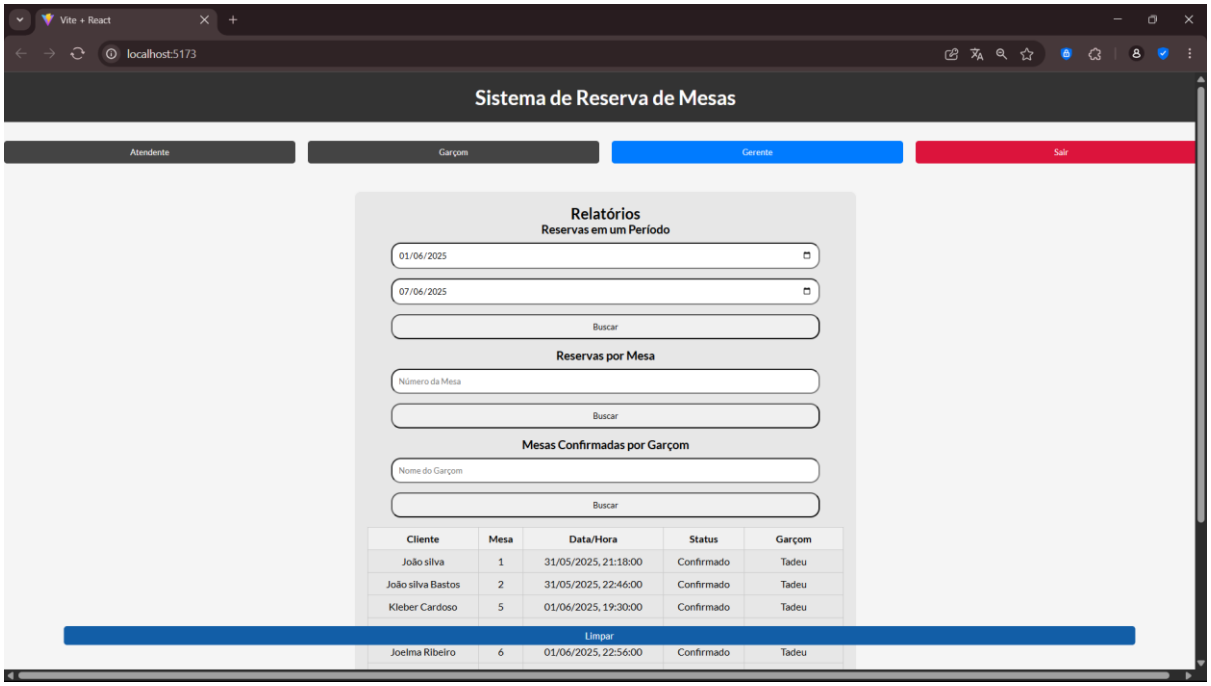
Depois do usuário preencher corretamente um ou mais dos campos os quais ele quer ver os dados e clicar no botão “Buscar” daquele campo aparecerá uma tabela com cinco colunas:

- Cliente: Exibindo o nome dos clientes.
- Mesa: Mostrando a mesa escolhida.
- Data/Hora: Exibindo o dia e a hora da reserva.
- Status: Mostrando a situação da reserva.
- Garçom: Mostrando o nome do garçom que confirmou a reserva.

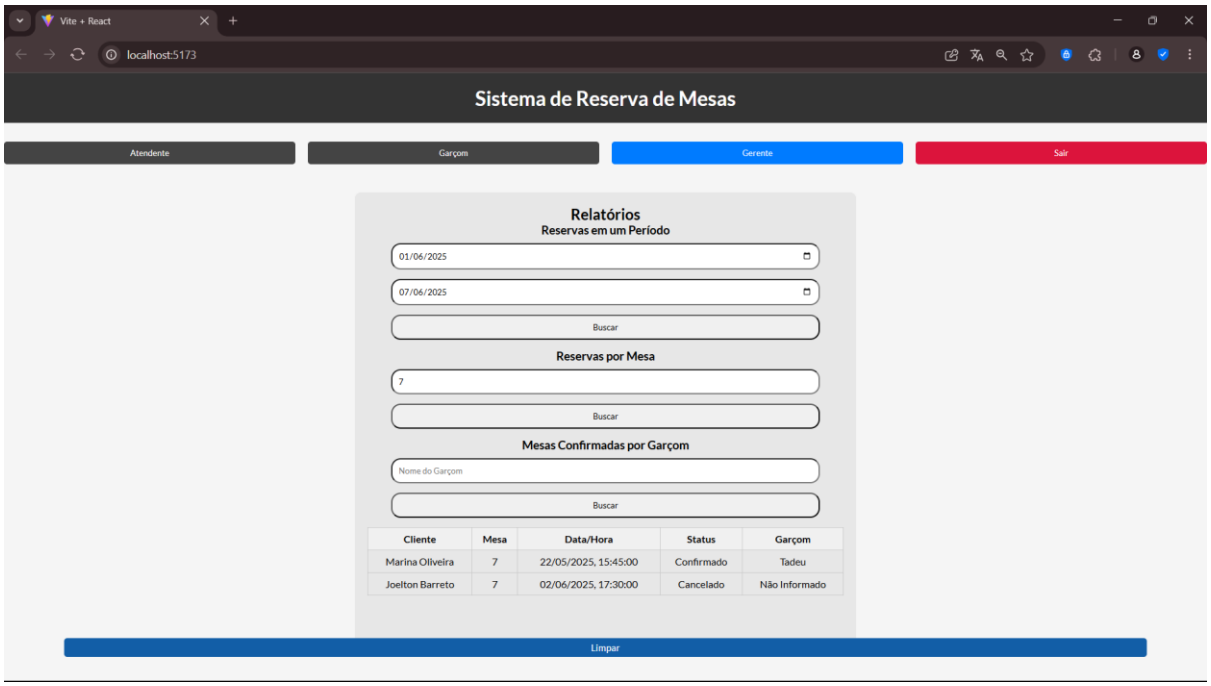
Caso o status do pedido esteja “cancelado”, quer dizer que o atendente cancelou o pedido e caso esteja “pendente”, quer dizer que nenhum garçom o confirmou ainda, em ambos os casos, a coluna garçom da tabela estará como “Não Informado”. Também há a opção de limpar os dados inseridos em todos os campos de texto e apagando a tabela com o botão “Limpar”.

Um pequeno detalhe é que ao preencher um dos campos e apertar o botão “Buscar”, a tabela aparecerá com os dados pedidos daquele campo, e se outro campo estiver preenchido, ao apertar o botão “buscar” dele, a tabela se atualizara com os novos dados pedidos. Por exemplo:

Aqui o relatório de reservas em um período foi preenchido, em seguida, ao apertar o botão “Buscar”, a tabela apareceu com os dados que foram pedidos.



Logo em seguida, o relatório de reservas por mesa foi preenchido e ao apertar o botão “Buscar”, a tabela atualizou com os novos dados que foram pedidos.



Sobre a Comunicação Usada

A utilização da API é essencial pois com ela é possível a conexão de diferentes aplicações, como o VSCODE com o NODE.JS, consequentemente trazendo mais agilidade e facilidade para uma série de processos, como a integração entre banco de dados, por exemplo.

Além disso, a segurança de dados é outro importante benefício das APIs, uma vez que elas são capazes de bloquear acessos e permissões a informações de softwares e hardwares, como um sistema de login e senha que utilizamos, por exemplo.