

Sistema de Gestão de Restaurantes

Este é um sistema de gestão de restaurantes desenvolvido com Node.js, Express, Sequelize, PostgreSQL e React.

Instalação

Pré-requisitos

- Node.js
- PostgreSQL
- npm

Banco de Dados

Criação do banco de dados

```
CREATE DATABASE erp;
```

```
CREATE TABLE Usuarios (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE NOT NULL,  
  senha VARCHAR(255) NOT NULL  
);
```

Backend

Navegue até a pasta do backend:

```
cd backend
```

Instale as dependências:

```
npm init -y
```

```
npm install express body-parser cors sequelize pg jwt-simple bcryptjs  
npm install
```

Frontend

Navegue até a pasta do frontend:

```
cd frontend
```

Instale as dependências:

```
npx create-react-app frontend
```

Configuração

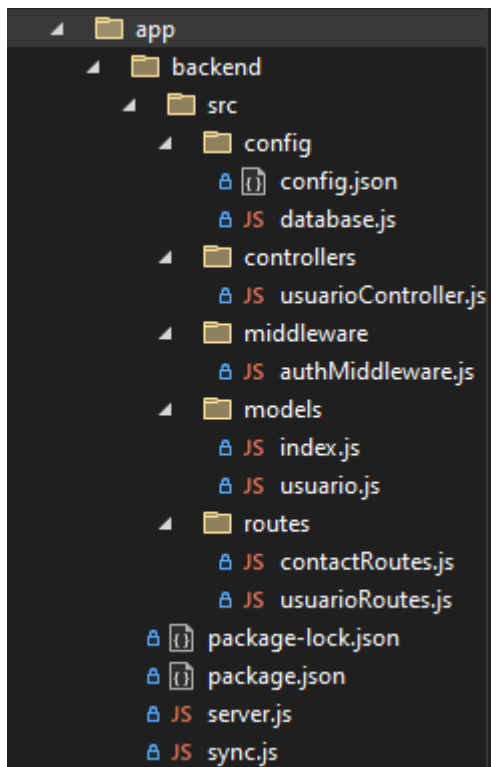
Banco de Dados

Configuração config.json:

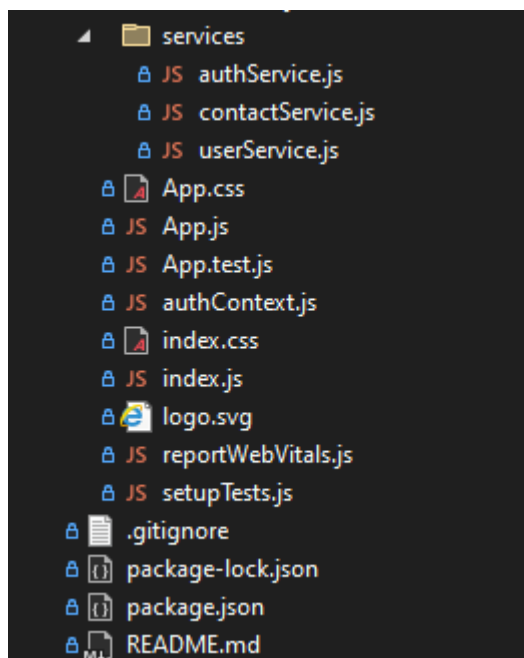
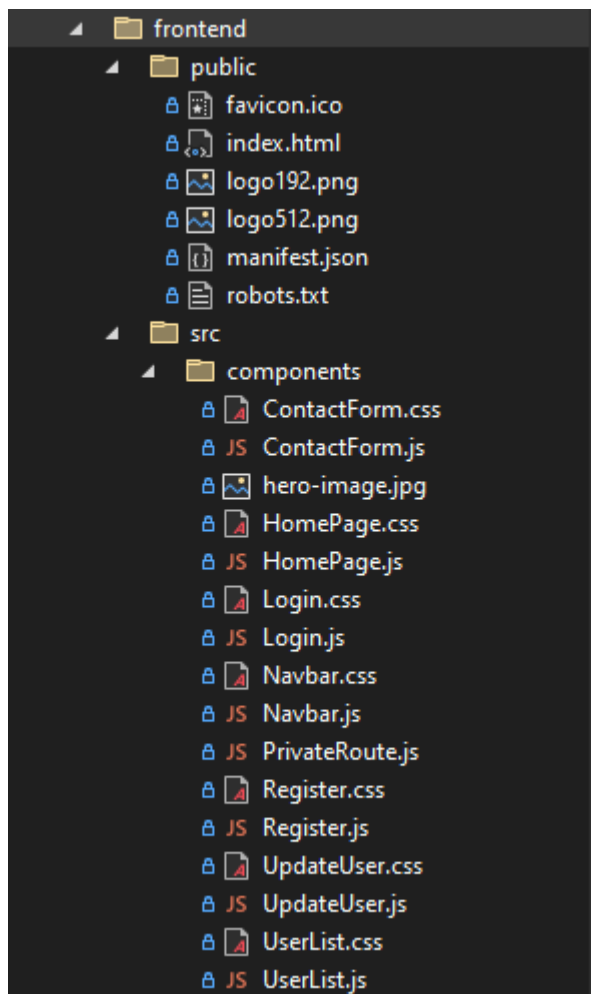
```
{
  "development": {
    "username": "postgres",
    "password": "admin",
    "database": "erp",
    "host": "localhost",
    "dialect": "postgres"
  }
}
```

Estrutura do Projeto

Backend



Frontend



Funcionalidades

Backend

- Gerenciamento de Usuários
 - Registro de Usuário
 - Login de Usuário
 - Atualização de Usuário
 - Exclusão de Usuário
 - Listagem de Usuários
 - Obtenção de Usuário por ID

Frontend

- Autenticação
 - Login
 - Registro
- Funcionalidades Protegidas
 - Listagem de Usuários (somente autenticado)
 - Atualização de Usuários
- Formulário de Contato

Uso

Backend

Para iniciar o servidor backend:

```
node server.js
```

Frontend

Para iniciar o servidor frontend:

```
npm start
```

A aplicação estará disponível em <http://localhost:3000> para o backend e <http://localhost:3001> para o frontend.

API

Endpoints

Autenticação

- POST /api/login: Autenticar um usuário.

Usuários

- POST /api/usuarios: Criar um novo usuário.
- GET /api/usuarios: Listar todos os usuários.
- GET /api/usuarios/:id: Obter um usuário por ID.
- PUT /api/usuarios/:id: Atualizar um usuário.
- DELETE /api/usuarios/:id: Excluir um usuário.

Frontend

Componentes

- **Navbar:** Barra de navegação do site.
- **HomePage:** Página inicial do site.
- **Login:** Página de login.
- **Register:** Página de registro de usuário.
- **UpdateUser:** Página de atualização de usuário.
- **UserList:** Página de listagem de usuários.
- **ContactForm:** Formulário de contato.

Contexto de Autenticação

O contexto de autenticação (authContext.js) gerencia o estado de autenticação do usuário na aplicação frontend.

Serviços

- **authService.js:** Serviço para autenticação de usuário.
- **userService.js:** Serviço para operações relacionadas aos usuários.