

Login do Administrador

1. Banco de Dados

É necessária uma tabela para armazenar as informações dos administradores. Conforme o planejamento, esta tabela pode ser chamada de `restaurantes` ou `usuarios_admin`.

Tabela: `restaurantes`

```
CREATE TABLE restaurantes (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  nome TEXT NOT NULL,  
  email TEXT NOT NULL UNIQUE,  
  senha_hash TEXT NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

- **id**: Identificador único para cada administrador (Chave Primária).
- **nome**: Nome do restaurante ou do administrador.
- **email**: Utilizado para o login e deve ser único no sistema.
- **senha_hash**: **Importante**: A senha nunca deve ser salva em texto puro. Armazenar sempre um "hash" – uma versão criptografada e irreversível da senha.
- **created_at**: Data e hora de criação do registro.

2. Backend

Serão necessários dois endpoints principais, conforme definido no planejamento.

Endpoint 1: Cadastro de Administrador

- **URL**: `POST /auth/register`
- **Função**:
 1. Recebe os dados do formulário de cadastro (nome, email, senha).
 2. Verifica se o e-mail já existe no banco de dados para garantir que seja único. Em caso afirmativo, retorna um erro.
 3. Cria um **hash** da senha recebida.
 4. Salva o novo usuário na tabela `restaurantes` com a senha já criptografada (hash).
 5. Retorna uma mensagem de sucesso.

Endpoint 2: Login do Administrador

- **URL**: `POST /auth/login`
- **Função**:
 1. Recebe `email` e `senha` do formulário de login.
 2. Busca no banco de dados um usuário com o `email` correspondente. Se não encontrar, retorna um erro de credenciais inválidas.
 3. Compara o hash da senha enviada pelo usuário com o `senha_hash` salvo no banco de dados.
 4. **Se as senhas corresponderem**: Gera um **Token JWT (JSON Web Token)**. Este token funcionará como uma credencial digital segura e temporária.
 5. Retorna o Token JWT para o frontend.

6. **Se as senhas não corresponderem:** Retorna um erro indicando credenciais inválidas.

3. Frontend

Tela 1: Cadastro

- **Componentes:** Um formulário com os seguintes campos:
 - Nome completo
 - Nome do restaurante
 - Email (deve ser único)
 - Senha (com requisito de no mínimo 8 caracteres)
 - Confirmar senha
 - Um botão: "Criar conta"
- **Funcionalidade:** Ao submeter o formulário, os dados são enviados para o endpoint `POST /auth/register`. O sistema deve exibir uma mensagem de sucesso ou erro, conforme a resposta do backend.

Tela 2: Login

- **Componentes:** Um formulário com os seguintes campos:
 - Email
 - Senha
 - Um botão: "Entrar"
- **Funcionalidade:**
 1. Ao clicar em "Entrar", os dados são enviados para o endpoint `POST /auth/login`.
 2. **Em caso de sucesso:**
 - O backend retorna um Token JWT.
 - O frontend deve **salvar este token** (geralmente no `localStorage` do navegador ou em cookies).
 - O administrador é redirecionado para o painel principal (dashboard).
 3. **Para acessos futuros:** O token salvo deve ser incluído no cabeçalho de todas as requisições à API (`Authorization: Bearer {token}`) para validar a sessão do usuário.
 4. **Em caso de falha:** O frontend deve exibir uma mensagem de erro (ex: "E-mail ou senha incorretos").