

Explicação e fluxo do projeto

A Cupcake Store é uma aplicação web desenvolvida com uma arquitetura simples e organizada, dividida em duas grandes partes: o backend, que concentra toda a lógica da API, e o frontend, que reúne toda a interface visual utilizada pelo cliente e pelo administrador. No backend, o arquivo principal é o `server.js`, responsável por iniciar o servidor Node.js com Express e conectar o banco de dados SQLite. Esse banco, armazenado em `cupcake_store.db`, contém todas as tabelas da aplicação, incluindo usuários, produtos, pedidos, cupons e itens do carrinho. Cada funcionalidade é separada em arquivos de rota dentro da pasta `routes`, o que facilita a manutenção. Assim, rotas de produtos, carrinho, pedidos, cupons e usuários ficam independentes umas das outras, enquanto o arquivo `admin.js` reúne endpoints específicos do painel administrativo.

Do outro lado, o frontend funciona como a camada visual que o usuário realmente enxerga. Ele está dividido em três partes principais: os arquivos HTML que representam cada tela, as folhas de estilo CSS responsáveis pelo visual e os scripts JavaScript que dão vida ao comportamento das páginas. As telas da loja ficam em `frontend/html`, onde estão páginas como `home`, `cardápio`, `carrinho`, `favoritos`, `histórico`, `checkout`, `login` e `perfil`. Já o painel administrativo possui sua própria área interna, também em HTML, com páginas para gerenciar produtos, usuários, cupons, pedidos e estatísticas da loja.

Cada página carrega seu próprio script, localizado na pasta `frontend/js`. Esses arquivos controlam tudo que o usuário faz na interface: carregar produtos do banco, adicionar itens ao carrinho, aplicar cupons, atualizar quantidades, exibir histórico de compras, gerenciar favoritos e processar a finalização do pedido. O painel admin também possui seus próprios scripts dentro de `frontend/js/admin`, responsáveis por atualizar produtos, salvar alterações, excluir itens do catálogo, criar cupons, acompanhar pedidos e alterar o status de itens ativos ou inativos.

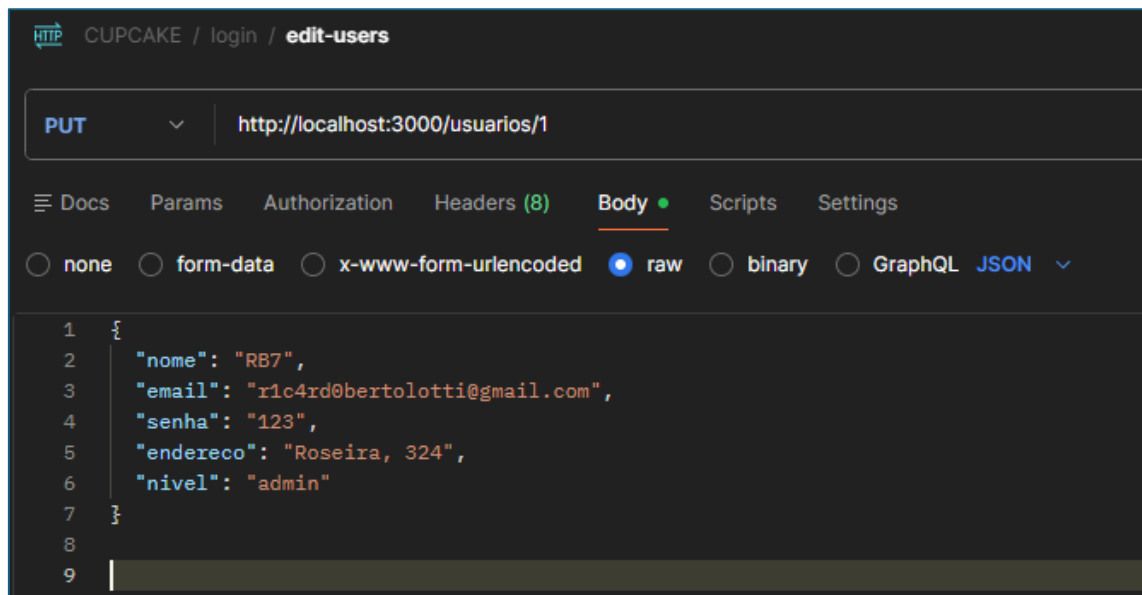
O fluxo completo funciona de forma bem simples: o frontend faz requisições HTTP para as rotas do backend, que respondem com dados reais do banco. Quando o usuário abre o cardápio, por exemplo, a página envia uma requisição para `/produtos` e recebe de volta a lista completa de cupcakes. Ao clicar em “adicionar ao carrinho”, outra requisição é enviada para `/carrinho`, registrando a

ação no banco de dados. Tudo segue esse padrão: o frontend pede, o backend responde e o banco grava ou retorna informações.

Por trás disso, imagens, ícones e ilustrações são mantidos na pasta `assets/img`, enquanto os estilos gerais da loja e do painel admin ficam separados em `style.css` e `style-admin.css`, garantindo organização visual. Finalmente, toda a parte de dependências do Node fica em `node_modules`, instalada automaticamente pelo `npm`, e os arquivos `package.json` e `package-lock.json` descrevem todas as bibliotecas usadas no projeto.

Assim, a Cupcake Store se estrutura como uma aplicação completa, visualmente simples, mas tecnicamente organizada, com divisão clara entre camadas, rotas bem definidas, banco funcional e fluxo lógico de comunicação entre cliente e servidor.

Exemplo de uma rota na API:



The screenshot shows a REST client interface with the following details:

- Method: **PUT**
- URL: `http://localhost:3000/usuarios/1`
- Body Type: **raw** (selected), with **JSON** as the content type.
- Body Content (JSON):

```
1 {  
2   "nome": "RB7",  
3   "email": "r1c4rd0bertolotti@gmail.com",  
4   "senha": "123",  
5   "endereco": "Roseira, 324",  
6   "nivel": "admin"  
7 }  
8  
9
```