# Documentação: Comunicação Frontend ↔ Backend

## Tecnologias Utilizadas

**Frontend:** Vue 3 + Vite + Pinia + Axios  
**Backend:** Node.js + Express + MongoDB

## Como Funciona a Comunicação

### 1. ****Configuração da API (Frontend)****

javascript

*// src/config/api.js*

import axios from 'axios';

const api = axios.create({

baseURL: 'http://localhost:3000/api'

});

* Axios cria uma instância configurada para se comunicar com o backend
* Todas as requisições partem de http://localhost:3000/api

### 2. ****Store Pinia (Gerenciamento de Estado)****

javascript

*// src/stores/userStore.js*

export const useUserStore = defineStore('user', {

actions: {

async fetchUsers() {

const response = await api.get('/users');

this.users = response.data;

}

}

});

* Pinia centraliza as chamadas à API
* Armazena os dados recebidos no estado da aplicação

### 3. ****Rotas do Backend (Express)****

javascript

*// backend/src/router/userRouter.js*

router.get('/users', async (req, res) => {

const users = await User.find();

res.json(users);

});

* Express recebe a requisição HTTP
* Busca dados no MongoDB
* Retorna JSON para o frontend

### 4. ****Fluxo Completo (Exemplo: Listar Usuários)****

1. Componente Vue chama: userStore.fetchUsers()

↓

2. Store faz requisição: api.get('/users')

↓

3. Axios envia: GET http://localhost:3000/api/users

↓

4. Express recebe em: router.get('/users', ...)

↓

5. Busca no MongoDB: User.find()

↓

6. Retorna JSON: res.json(users)

↓

7. Axios recebe response.data

↓

8. Store atualiza: this.users = response.data

↓

9. Componente Vue renderiza a lista

## Operações CRUD

| **Operação** | **Frontend** | **Backend** | **MongoDB** |
| --- | --- | --- | --- |
| **Listar** | api.get('/users') | GET /api/users | User.find() |
| **Criar** | api.post('/users', dados) | POST /api/users | User.create() |
| **Atualizar** | api.put('/users/:id', dados) | PUT /api/users/:id | User.findByIdAndUpdate() |
| **Deletar** | api.delete('/users/:id') | DELETE /api/users/:id | User.findByIdAndDelete() |

## Resumo

1. **Vue** → chama métodos da **Store Pinia**
2. **Pinia** → usa **Axios** para fazer requisições HTTP
3. **Axios** → envia requisições para o **Express**
4. **Express** → processa e consulta o **MongoDB**
5. **MongoDB** → retorna dados para o **Express**
6. **Express** → envia resposta JSON
7. **Axios** → recebe e repassa para a **Store**
8. **Store** → atualiza estado
9. **Vue** → renderiza interface com dados atualizados

**Comunicação:** Frontend e Backend se comunicam via **requisições HTTP REST** usando formato **JSON**.