

Criando um CRUD Simples com HTML, JavaScript, Node.js e SQLite

1. Criando a Estrutura do Projeto

Abra um terminal e execute os seguintes comandos para criar a pasta do projeto e acessar ela:

```
mkdir cadastro-usuarios  
cd cadastro-usuarios
```

Agora, crie as pastas para separar o frontend e backend:

```
mkdir backend frontend
```

2. Configurando o Servidor (Backend) com Node.js e Express

Entre na pasta `backend`:

```
cd backend
```

2.1 Criar o arquivo de configuração do Node.js

Execute o comando abaixo para iniciar um novo projeto Node.js e gerar o arquivo `package.json`:

```
npm init -y
```

Isso criará um `package.json` contendo as configurações básicas do projeto.

2.2 Instalar as dependências

Agora, instale os pacotes necessários:

```
npm install express sqlite3 cors
```

- **express** → Framework para criar o servidor web.
- **sqlite3** → Biblioteca para gerenciar o banco de dados SQLite.
- **cors** → Permite que o frontend se comunique com o backend sem bloqueios.

2.3 Criar o arquivo do servidor

Agora, crie o arquivo principal do backend:

```
type nul > server.js
```

Abra o arquivo `server.js` e adicione o seguinte código:

```
const express = require("express");
const sqlite3 = require("sqlite3").verbose();
const cors = require("cors");

const app = express();
app.use(express.json());
app.use(cors());

// Conectar ao banco de dados SQLite
const db = new sqlite3.Database("users.db", (err) => {
  if (err) return console.error(err.message);
  console.log("Conectado ao banco de dados SQLite.");
});

// Criar tabela de usuários se não existir
db.run("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, name TEXT, email TEXT)");

// Rota para cadastrar um usuário
app.post("/users", (req, res) => {
  const { name, email } = req.body;
  db.run("INSERT INTO users (name, email) VALUES (?, ?)", [name, email],
function(err) {
  if (err) return res.status(500).json({ error: err.message });
  res.json({ id: this.lastID });
});
});

// Rota para listar todos os usuários
app.get("/users", (req, res) => {
  db.all("SELECT * FROM users", [], (err, rows) => {
    if (err) return res.status(500).json({ error: err.message });
    res.json(rows);
  });
});

// Iniciar o servidor na porta 3000
app.listen(3000, () => console.log("Servidor rodando na porta 3000"));
```

Agora, execute o servidor:

```
node server.js
```

Se tudo estiver correto, você verá a mensagem:

Servidor rodando na porta 3000

🔗 3. Criando o Frontend (HTML + JavaScript)

Agora, volte para a pasta raiz e acesse a pasta `frontend`:

```
cd ../frontend
```

Crie um arquivo `index.html`:

```
type nul > index.html
```

Abra o `index.html` e cole o seguinte código:

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cadastro de Usuários</title>
</head>
<body>
  <h2>Cadastro de Usuários</h2>
  <form id="userForm">
    <label for="name">Nome:</label>
    <input type="text" id="name" required>
    <label for="email">E-mail:</label>
    <input type="email" id="email" required>
    <button type="submit">Cadastrar</button>
  </form>

  <h2>Lista de Usuários</h2>
  <table border="1">
    <thead>
      <tr>
        <th>ID</th>
        <th>Nome</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody id="userTable"></tbody>
  </table>

  <script>
    // Enviar dados para o backend
    document.getElementById("userForm").addEventListener("submit", async
function(event) {
  event.preventDefault();
  const name = document.getElementById("name").value;
  const email = document.getElementById("email").value;

  const response = await fetch("http://localhost:3000/users", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ name, email })
  });
});
```

```

        if (response.ok) {
            document.getElementById("userForm").reset();
            loadUsers();
        }
    });

    // Carregar usuários do backend
    async function loadUsers() {
        const response = await fetch("http://localhost:3000/users");
        const users = await response.json();
        const userTable = document.getElementById("userTable");
        userTable.innerHTML = "";
        users.forEach(user => {
            const row =
`<tr><td>${user.id}</td><td>${user.name}</td><td>${user.email}</td></tr>`;
            userTable.innerHTML += row;
        });
    }

    loadUsers(); // Carrega os usuários ao iniciar a página
</script>
</body>
</html>

```

🚩 4. Executando o Projeto

4.1 Iniciar o Backend

Volte para a pasta `backend` e execute:

```
node server.js
```

Se tudo estiver certo, o servidor estará rodando na porta 3000.

4.2 Abrir o Frontend

Agora, abra o arquivo `index.html` no seu navegador e teste a aplicação!

🔥 Resumo do Projeto

📁 cadastro-usuarios/

