

Apresentação

Joalheria

Raí Costa



Bach - Emma

Passo 1: Configuração Inicial

1. Criar e inicializar o projeto:

- Crie uma pasta para o projeto e rode npm init -y para inicializar o package.json.

2. Instalar dependências:

- Instale as dependências essenciais: express, mongoose, cors e dotenv:
 - npm install express mongoose cors dotenv

Passo 2: Estrutura de Pastas

```
loja-joalheria/
|
├── controllers/          # Controladores (Lógica das rotas)
├── models/                # Modelos de dados
├── routes/                # Rotas da API
├── config/                # Configurações gerais
├── .env                    # Variáveis de ambiente
└── app.js                 # Arquivo principal do servidor
```

Passo 3: Configuração do Servidor Express

```
const express = require('express');
const cors = require('cors');
const mongoose = require('mongoose');
const dotenv = require('dotenv');

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

// Conectar ao MongoDB
mongoose.connect(process.env.MONGO_URI)
  .then(() => console.log('Conectado ao MongoDB'))
  .catch(err => console.log('Erro de conexão: ', err));

app.listen(process.env.PORT || 5000, () => {
  console.log(`Servidor rodando na porta ${process.env.PORT || 5000}`);
});
```

Passo 4: Modelos de Dados (Mongoose)

4.1. Modelo de Produto (exemplo: Produto.js):

```
const mongoose = require('mongoose');

const produtoSchema = new mongoose.Schema({
    nome: String,
    descricao: String,
    preco: Number,
    imagem: String
});

module.exports = mongoose.model('Produto', produtoSchema);
```

Passo 4: Modelos de Dados (Mongoose)

4.2. Modelo de Cliente (exemplo: Cliente.js):

```
const mongoose = require('mongoose');

const clienteSchema = new mongoose.Schema({
    nome: String,
    email: String,
    telefone: String,
    endereco: String
});

module.exports = mongoose.model('Cliente', clienteSchema);
```

Passo 5: Controladores de CRUD

```
const Produto = require('../models/Produto');

// Criar produto
exports.criarProduto = async (req, res) => {
    const produto = new Produto(req.body);
    await produto.save();
    res.status(201).json(produto);
};

// Obter todos os produtos
exports.getProdutos = async (req, res) => {
    const produtos = await Produto.find();
    res.status(200).json(produtos);
};
```

Passo 6: Rotas da API

```
const express = require('express');
const router = express.Router();
const produtoController = require('../controllers/produtoController');

// Rotas de CRUD para Produto
router.post('/', produtoController.criarProduto);
router.get('/', produtoController.getProdutos);

module.exports = router;
```

Passo 7: Variáveis de Ambiente

```
MONGO_URI=mongodb://localhost:27017/loja_joias  
PORT=5000
```

Passo 8: Testar a API

Após configurar as rotas, controladores e modelos, teste a API utilizando ferramentas como Postman ou Insomnia para realizar requisições GET, POST, PUT e DELETE.



*Obrigado pela
Atenção*

