

# **BACKEND COM NODE.JS**

## **AULA 8-9-10 – PROJETO FINAL**

Por **Wanderson Guimarães – Senac Lapa Tito**

# Sobre o nosso sistema final

Terá quatro chamadas: register , login , categorias, pedidos e produtos.

Usará **Express** para a API,

**Mongoose** para o MongoDB

**JWT** para autenticação

**Bcrypt** para hash de senhas

**Dotenv** para variáveis de ambiente

# Estrutura do Projeto

```
market-api/  
├─ config/  
│   └─ db.js  
├─ controllers/  
│   ├── auth.controller.js  
│   ├── product.controller.js  
│   ├── category.controller.js  
│   └─ order.controller.js  
├─ middleware/  
│   └─ auth.middleware.js  
├─ models/  
│   ├── user.model.js  
│   ├── product.model.js  
│   ├── category.model.js  
│   └─ order.model.js  
├─ routes/  
│   ├── auth.routes.js  
│   ├── product.routes.js  
│   ├── category.routes.js  
│   └─ order.routes.js  
├─ .env  
├─ package.json  
└─ server.js
```

# O que é MongoDB?

- Banco de dados **NoSQL** baseado em documentos .
- Utiliza o formato **JSON-like** (BSON) .
- Flexível, escalável e schema-less.
- Muito usado em aplicações modernas (Node.js, MERN stack).

# Mongo Db – Principais Conceitos

| Termo<br>MongoDB | Equivalente Banco<br>Relacional |
|------------------|---------------------------------|
| Database         | Banco de Dados                  |
| Collection       | Tabela                          |
| Document         | Linha (Registro)                |
| Field            | Coluna                          |

# Mongo Db – O que é uma Collection?

- Conjunto de documentos
- Criada automaticamente ao inserir o primeiro documento
- Não exige esquema fixo
- Suporta indexação, filtros e agregações



# Mongo Db – O que é Mongoose

ODM (Object Data Modeling) para **Node.js + MongoDB**

- Facilita:
- Criação de modelos com **schemas**
- Validação de dados
- Relacionamentos
- Middleware (pre/post hooks)

# Mongo Db – Como usar o MongoDB Atlas (Free Tier)

- Acesse <https://www.mongodb.com/cloud/atlas>
- Crie uma conta gratuita
- Crie um cluster (Shared Tier)
- Configure um **usuário de acesso**
- Adicione seu IP na **lista de IPs permitidos**
- Conecte usando o link fornecido



# Mongo Db – Boas Práticas

- Valide dados com Mongoose
- Use índices para performance
- Faça backups (mesmo no plano gratuito)
- Use variáveis de ambiente para a conexão (.env)

# **Mongo Db – Instalando o MongoDB Localmente**

- **Pré-requisitos**
- **Sistema: Windows, Linux ou macOS**
- **Espaço em disco (~300 MB)**
- **Acesso de administrador**

# Mongo Db – Instalação Windows

- Acesse: [mongodb.com/try/download/community](https://mongodb.com/try/download/community)
- Baixe o instalador .msi
- Execute o instalador:
- Marque "Install MongoDB as a Service"
- Instale também o MongoDB Compass (opcional)
- Finalize e abra o terminal

# Mongo Db – Instalação Linux

```
sudo apt update
```

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc
```

```
| sudo apt-key add -
```

```
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu focal/
```

```
mongodb-org/6.0 multiverse" | sudo tee /
```

```
etc/apt/sources.list.d/mongodb-org-6.0.list
```

```
sudo apt update
```

```
sudo apt install -y mongodb-org
```

## **Mongo Db – Instalação Mac OS (HOMEBREW)**

- `brew tap mongodb/brew`
- `brew install mongodb-community@6.0`
- `brew services start mongodb-community@6.0`



# Mongo Db – Rodando o MongoDB

```
mongod --version  
mongo
```



# Mongo Db - Resumo

- MongoDB é um banco NoSQL baseado em documentos
- Collections são como tabelas, mas flexíveis
- Mongoose traz estrutura ao usar MongoDB com Node.js
- MongoDB Atlas permite usar MongoDB gratuitamente na nuvem

# Autenticação JWT com Node.js e MongoDB

**Geração e uso do Bearer Token**

# 1. Registro de Usuário

Método: POST /auth/register

Corpo da requisição:

```
{  
  "name": "Admin",  
  "email": "admin@example.com",  
  "password": "123456"  
}
```

## 2. Rota de Login

Método: POST /auth/login

Corpo da requisição (JSON):

```
{  
  "email": "admin@example.com",  
  "password": "123456"  
}
```

### 3. Resposta com Token

Se as credenciais estiverem corretas: Pegue o token gerado aqui e coloque aqui :

```
{  
  "token": "eyJhbGciOiJIUzI1NiIs..."  
}
```

## 4. Uso do Token Bearer

Enviar nas rotas protegidas:

Authorization: Bearer <token>

Exemplo de rotas protegidas:

- POST /products
- PUT /products/:id
- DELETE /orders/:id



## 5. Usando o Postman

- ◆ Registre um usuário (POST /auth/register)
- ◆ Faça login (POST /auth/login)
- ◆ Copie o token JWT
- ◆ Cole o token em Authorization > Bearer Token

## 6. Como usar o sistema:

- ◆ Com usuário logado crie uma categoria ex : frutas
- ◆ Liste as categorias.
- ◆ Crie um produto passando uma categoria : ex: bananas
- ◆ Liste os produtos.

**Chegamos ao fim do curso , Obrigado !**

Meus Contatos:

Linkedin : <https://br.linkedin.com/in/wanderson-guimaraes>