Centro Paula Souza

Fatec Jacareí
Prof. Francisco de Moura

BANCO DE DADOS NÃO RELACIONAL

Integração com Aplicações Externas e APIs (MongoDB + Node.js)

Professora: Lucineide Pimenta

Tópicos da Aula

Fatec
Jacareí
Prof. Francisco de Moura

O que vamos aprender:

- Entender como conectar o MongoDB a uma aplicação Node.js.
- Criar uma API REST que manipule dados do MongoDB.
- Aplicar conceitos de CRUD via API.
- Preparar o ambiente para integração com o projeto ABP.

Conceito de Integração com APIs

Fatec
Jacareí
Prof. Francisco de Moura

- **API (Application Programming Interface):**
  Interface que permite que diferentes sistemas se comuniquem.
- No nosso caso:
  Aplicação Node.js ⇄ MongoDB Database.
- Vamos criar endpoints HTTP (rota) para **consultar**, **inserir**, **atualizar** e **remover** dados.

O que é uma API REST?

Fatec
Jacareí
Prof. Francisco de Moura

- **REST** = *Representational State Transfer*.
- Usa os métodos HTTP para manipular recursos:
  - **GET** → ler dados
  - **POST** → criar dados
  - **PUT** → atualizar dados
  - **DELETE** → excluir dados

Estrutura de uma API com Node.js

Fatec
Jacareí
Prof. Francisco de Moura

- **Projeto típico:**

```
/projeto-api
  ├── server.js
  ├── package.json
  ├── /routes
  ├── /models
  └──── /controllers
```

Preparando o Ambiente

Fatec
Jacareí
Prof. Francisco de Moura

1-Instalar Node.js (versão LTS).

2-Criar o diretório do projeto:

mkdir api-meteorologica

cd api-meteorologica

3-Inicializar o projeto:

npm init -y

4-Instalar dependências:

npm install express mongoose nodemon cors

Conectando ao MongoDB

Fatec
Jacareí
Prof. Francisco de Moura

Arquivo: server.js

```
const express = require("express");
const mongoose = require("mongoose");
const app = express();

app.use(express.json());

// Conexão com o MongoDB local
mongoose.connect("mongodb://127.0.0.1:27017/estacao_meteorologica")
  .then(() => console.log("Conectado ao MongoDB"))
  .catch(err => console.error("Erro ao conectar", err));

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

Criando o Modelo (Schema)

Fatec
Jacareí
Prof. Francisco de Moura

Arquivo: models/Leitura.js

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

const leituraSchema = new mongoose.Schema({
  cidade: String,
  sensor: String,
  valor: Number,
  data: { type: Date, default: Date.now }
});

module.exports = mongoose.model("Leitura", leituraSchema);
```

Criando as Rotas da API

Fatec
Jacareí
Prof. Francisco de Moura

Arquivo: routes/leituras.js

```
const express = require("express");
const router = express.Router();
const Leitura = require("../models/Leitura");

// Listar todas as leituras
router.get("/", async (req, res) => {
  const leituras = await Leitura.find();
  res.json(leituras);
});

// Inserir nova leitura
router.post("/", async (req, res) => {
  const leitura = new Leitura(req.body);
  await leitura.save();
  res.status(201).json(leitura);
});

module.exports = router;
```

Conectando as Rotas ao Servidor

Fatec
Jacareí
Prof. Francisco de Moura

- **Arquivo:** server.js

```
const leituraRoutes = require("./routes/leituras");
app.use("/leituras", leituraRoutes);
```

- **Agora nossa API responde em:**
  GET http://localhost:3000/leituras
  POST http://localhost:3000/leituras

Testando a API

- Use o **Postman** ou o **Insomnia** para testar.
- **Método GET** → lista leituras.
- **Método POST** → cria nova leitura.
- **Exemplo de corpo JSON**:

```
{
  "cidade": "Recife",
  "sensor": "temperatura",
  "valor": 33
}
```

Implementando Atualização (PUT)

Fatec
Jacareí
Prof. Francisco de Moura

```
// Atualizar leitura por ID
router.put("/:id", async (req, res) => {
  const leituraAtualizada = await
Leitura.findByIdAndUpdate(req.params.id, req.body, { new: true });
  res.json(leituraAtualizada);
});
```

Implementando Exclusão (DELETE)

Fatec
Jacareí
Prof. Francisco de Moura

```
router.delete("/:id", async (req, res) => {
  await Leitura.findByIdAndDelete(req.params.id);
  res.json({ mensagem: "Leitura removida com sucesso!" });
});
```

Estrutura CRUD Completa

Fatec
Jacareí
Prof. Francisco de Moura

Método	Caminho	Ação	Exemplo
GET	/leituras	Listar leituras	Listar todas
POST	/leituras	Criar leitura	Inserir JSON
PUT	/leituras/:id	Atualizar leitura	Atualizar valor
DELETE	/leituras/:id	Remover leitura	Deletar por ID

Integração com Dados Meteorológicos

Fatec
Jacareí
Prof. Francisco de Moura

- **Para integrar com os dados do projeto ABP:**
- **Coleção leituras deve conter sensores:** temperatura, umidade, vento, pressão.
- É possível consumir uma **API externa** (por exemplo, OpenWeatherMap) e **gravar** no MongoDB.

Exemplo de Consumo de API Externa

Fatec Jacareí
Prof. Francisco de Moura

```
const axios = require("axios");

async function importarDados() {
  const resposta = await
axios.get("https://api.openweathermap.org/data/2.5/weather?q=Fortaleza&appid=SUA_CHAVE");
  const dados = {
    cidade: resposta.data.name,
    sensor: "temperatura",
    valor: resposta.data.main.temp
  };
  await Leitura.create(dados);
}
```

Executando a Função

Fatec
Jacareí
Prof. Francisco de Moura

- Chamar **importarDados()** para coletar e armazenar dados automaticamente.
- Pode ser agendado com **node-cron** para rodar a cada hora.

Comparando com PostgreSQL

Fatec
Jacareí
Prof. Francisco de Moura

- **No SQL, seria algo como:**

INSERT INTO leituras (cidade, sensor, valor, data)

VALUES ('Recife', 'temperatura', 33, CURRENT_DATE);

- **No MongoDB usamos um documento JSON:**

db.leituras.insertOne({ cidade: "Recife", sensor: "temperatura", valor: 33 });

Boas Práticas na API

Fatec
Jacareí
Prof. Francisco de Moura

- Validar dados antes de inserir.
- Retornar mensagens claras no formato JSON.
- Utilizar códigos HTTP adequados (200, 201, 400, 404).
- Implementar tratamento de erros com **try/catch**.

Tratamento de Erros

Fatec
Jacareí
Prof. Francisco de Moura

```
router.get("/", async (req, res) => {
  try {
    const leituras = await Leitura.find();
    res.json(leituras);
  } catch (erro) {
    res.status(500).json({ erro: "Erro ao buscar leituras" });
  }
});
```

Autenticação (conceito)

Fatec
Jacareí
Prof. Francisco de Moura

- Em APIs reais, usamos **JWT (JSON Web Token)** para proteger rotas.
- **Exemplo:** apenas usuários logados podem criar leituras.

Revisão da Aula

Fatec
Jacareí
Prof. Francisco de Moura

- Conexão **MongoDB + Node.js**
- Criação de **API REST**
- Consumo de **API** externa
- **CRUD** completo via **API**

Boas Práticas

Fatec
Jacareí
Prof. Francisco de Moura

- Usar **async/await**.
- Dividir responsabilidades (rotas, modelos, controllers).
- Documentar **endpoints**.

Dicas Extras

Fatec
Jacareí
Prof. Francisco de Moura

- Usar **nodemon** para atualizar o servidor automaticamente.
- Usar **dotenv** para esconder credenciais.
- Criar um arquivo **.env** com a **string** de conexão.

Revisão Geral

Fatec
Jacareí
Prof. Francisco de Moura

- API REST usa métodos HTTP.
- MongoDB armazena documentos JSON.
- Node.js faz a ponte entre cliente e banco.

BANCO DE DADOS NÃO RELACIONAL

Atividade Prática (Individual)

Atividade Prática (Individual)

Fatec
Jacareí
Prof. Francisco de Moura

1. Criar o banco api_games.
2. Criar uma coleção jogos com campos: nome, categoria, preco, estoque.
3. Implementar uma API com rotas:
    1. GET /jogos
    2. POST /jogos
    3. PUT /jogos/:id
    4. DELETE /jogos/:id
4. Testar com Postman.
5. Inserir print de cada operação no Word.

Referências Bibliográficas

- **Material de apoio:**
- Chodorow, Kristina. *MongoDB: The Definitive Guide*. O'Reilly Media, 2013.
- *PostgreSQL Documentation*. Disponível em: https://www.postgresql.org/docs/
- *MongoDB Documentation*. Disponível em: https://www.mongodb.com/docs/
- Cattell, Rick. *Scalable SQL and NoSQL Data Stores*. ACM, 2011.
- *Mais detalhes sobre operadores:*
  https://www.mongodb.com/docs/manual/reference/operator/query
- *Operadores de atualização:*
  (https://www.mongodb.com/docs/manual/reference/operator/update/#update-operators-1)
- Documentação MongoDB CRUD
- MongoDB Aggregation Framework
- MongoDB Indexação e Performance

Bibliografia Básica

Fatec
Jacareí
Prof. Francisco de Moura

- BOAGLIO, Fernando**. MongoDB:** Construa novas aplicações com novas tecnologias. São Paulo: Casa do Código, 2015.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados:** Fundamentos e Aplicações. 7ed. São Paulo: Pearson, 2019.
- SADALAGE, P.; FOWLER, M. **Nosql Essencial:** Um Guia Conciso Para o Mundo Emergente da Persistência Poliglota. São Paulo: Novatec, 2013.
- SINGH, Harry. **Data Warehouse**: conceitos, tecnologias, implementação e gerenciamento. São Paulo: Makron Books, 2001.

Bibliografia Complementar

Fatec
Jacareí
Prof. Francisco de Moura

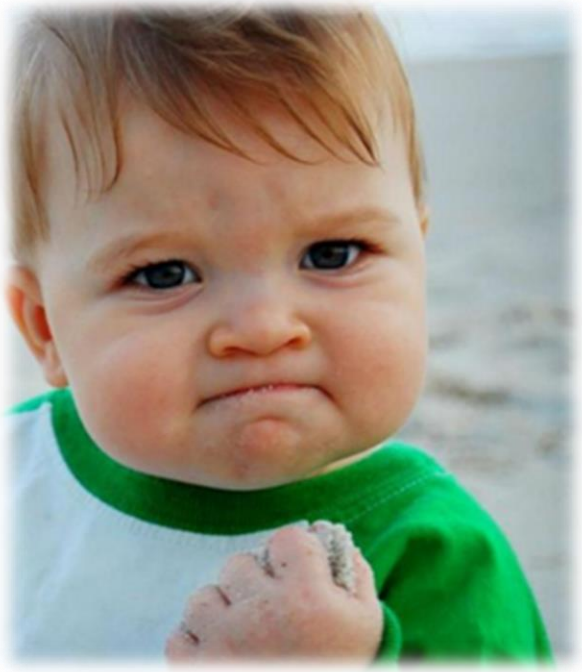
- FAROULT, Stephane. **Refatorando Aplicativos SQL**. Rio de Janeiro: Alta Books, 2009.
- PANIZ, D. **NoSQL:** Como armazenar os dados de uma aplicação moderna. Casa do Código, 2016.
- SOUZA, M. **Desvendando o Mongodb**. Rio de Janeiro: Ciência Moderna, 2015.

Dúvidas?

Fatec
Jacareí
Prof. Francisco de Moura

Considerações Finais

Fatec
Jacareí
Prof. Francisco de Moura



Professora:

Lucineide Pimenta

Bom descanso à todos!