

PRÁTICAS AVANÇADAS EM DESENVOLVIMENTO WEB

Davi Schneid - davi.Schneid@gmail.com

15/07/2024

Agenda

- ▶ Adicionar user e senha Mongo
- ▶ Criar .bat iniciar MongoDB
- ▶ Criar JS acesso ao MongoDB
 - ▶ Criar rota de cadastro
 - ▶ Criar rota de edição
 - ▶ Criar rota buscar por nome
- ▶ Revisão
- ▶ Gamification

Add user Mongo

- ▶ Baixar mongo Shell
<https://www.mongodb.com/try/download/shell>

- ▶ Executar mongoSH

- ▶ Informar URI-> **mongodb://127.0.0.1/27017**

- ▶ Executar comando: **use admin**

```
27017> use admin
switched to db admin
```

- ▶ Criar user ADMIN, executar comando:

```
admin> db.createUser({
...   user: "davi.schneid",
...   pwd: "12345678",
...   roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
... })
{ ok: 1 }
admin> db.auth("davi.schneid", "12345678")
{ ok: 1 }
```

- ▶ Criar user mongodb

```
admin> use mongodb
switched to db mongodb
mongodb> db.createUser({
...   user: "mongo",
...   pwd: "12345678",
...   roles: [ { role: "readWrite", db: "mongodb" } ]
... })
{ ok: 1 }
mongodb>
```

Add user Mongo

- ▶ Criar arquivo mongo.cfg na pasta BIN

```
PS C:\MongoDB\bin> ls

Directory: C:\MongoDB\bin

Mode                LastWriteTime         Length Name
----                -
-a---             09/07/2024   14:56           1558 Install-Compass.ps1
-a---             14/07/2024   17:23             35 mongod.cfg
-a---             09/07/2024   14:56        63604224 mongod.exe
-a---             09/07/2024   14:56       1032704000 mongod.pdb
-a---             09/07/2024   14:57        37261824 mongos.exe
-a---             09/07/2024   14:57       580923392 mongos.pdb
-a---             09/07/2024   14:57       25314320  vc_redist.x64.exe

PS C:\MongoDB\bin> pwd

Path
----
C:\MongoDB\bin
```

- ▶ Add parâmetros configuração

- ▶ Parâmetro storage
- ▶ Parâmetro segurança
- ▶ Parâmetro SystemLog

```
PowerShell 7 (x64)

PS C:\MongoDB\bin> more .\mongod.cfg
security:
  authorization: enabled

storage:
  dbPath: C:\MongoDB\DATA

systemLog:
  destination: file
  path: C:\MongoDB\LOG\mongod.log
  logAppend: true
PS C:\MongoDB\bin> _
```

- ▶ Reiniciar Mongoddb

```
Eric@DESKTOP-8CI3MGV MINGW64 /c/MongoDB/bin
$ ./mongod.exe --config ../bin/mongod.cfg
```

Mongo Compass

► Criar conexão banco

MongoDB Compass
Connect Edit View Help

Compass

New connection +

Saved connections

Recents

127.0.0.1
14 de jul. de 2024, 17:25

localhost:27017
14 de jul. de 2024, 17:51

New Connection

Connect to a MongoDB deployment



FAVORITE

URI ⓘ

Edit Connection String ☒

mongodb://mongo:*****@localhost:27017/?authSource=mongodb

Advanced Connection Options

General

Authentication

TLS/SSL

Proxy/SSH

In-Use Encryption

Advanced

Authentication Method

Username/Password

OIDC

X.509

Kerberos

LDAP

AWS IAM

Username

mongo

Optional

Password

Optional

Authentication Database ⓘ

mongodb

Optional

Authentication Mechanism

Default

SCRAM-SHA-1

SCRAM-SHA-256

Save

Save & Connect

Connect

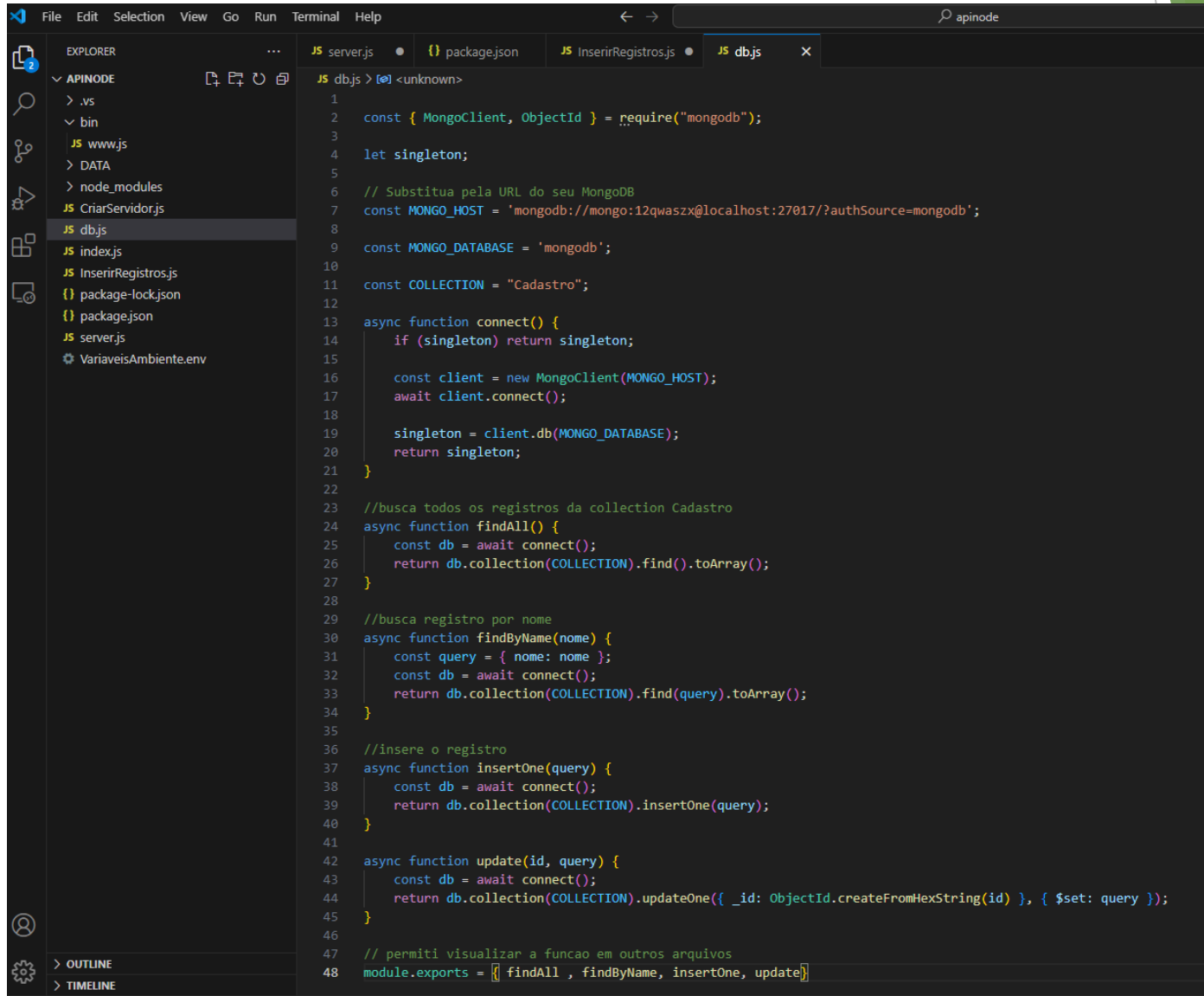
Criar .bat Iniciar MongoDB

- ▶ Criar pasta SCRIPTS dir:C:\MongoDB\SCRIPTS>
- ▶ Criar script StartMongoDB.bat

```
1  @echo off
2  setlocal enabledelayedexpansion
3
4  echo [Starting MongoDB...]
5  cd C:\MongoDB\bin
6
7  mongod.exe --config mongod.cfg
8
9  cd C:\MongoDB\SCRIPTS
10
11  :start
12  echo [MongoDB is running...]
13  cd C:\MongoDB\SCRIPTS
14  timeout /t 300 /nobreak
15  goto start
```

Buscar dados banco

► Criar arquivo db.js

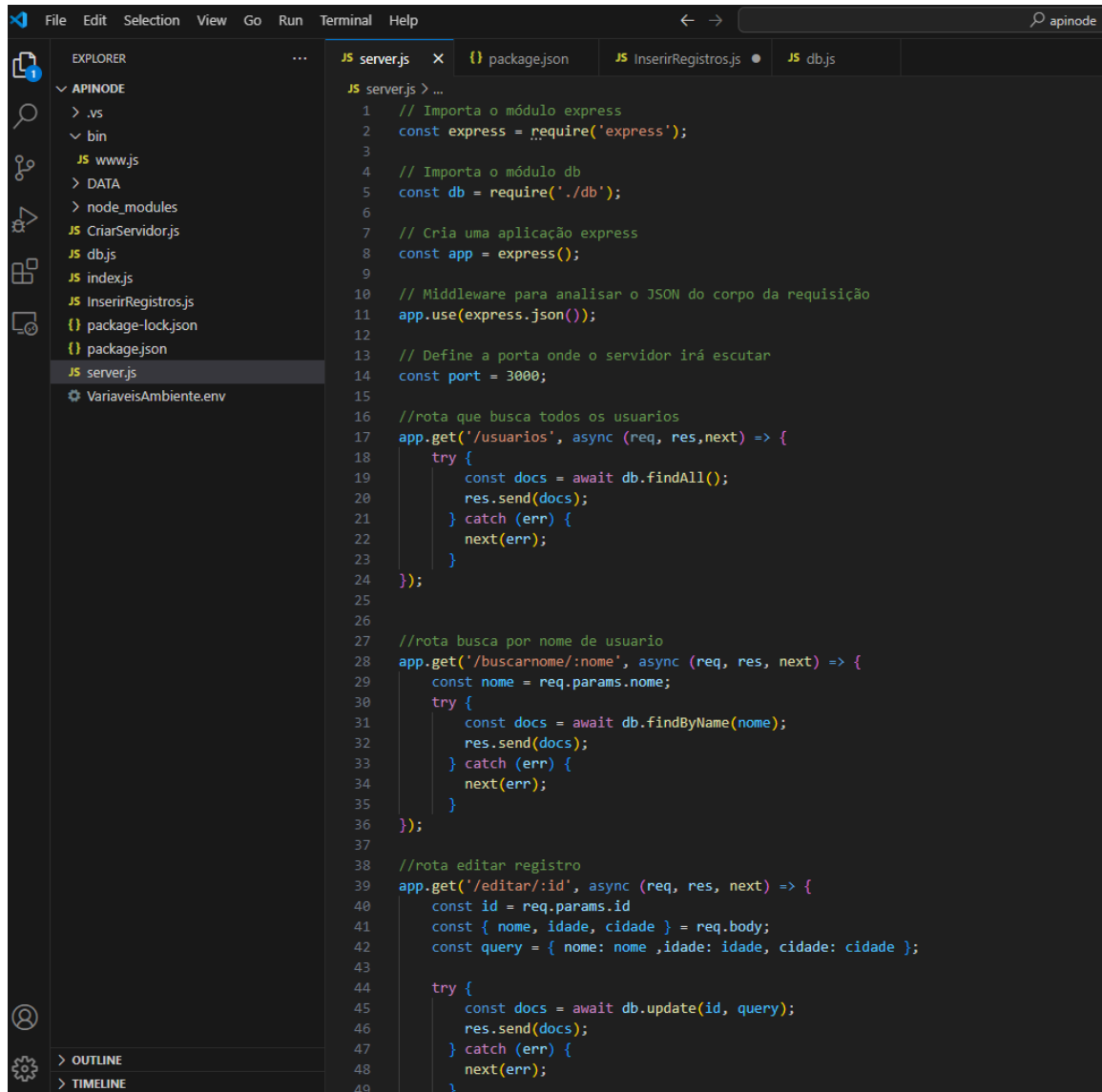


The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays the project structure for 'APINODE', including files like 'server.js', 'package.json', 'InserirRegistros.js', and the newly created 'db.js'. The main editor window shows the content of 'db.js', which contains the following code:

```
1
2  const { MongoClient, ObjectId } = require("mongodb");
3
4  let singleton;
5
6  // Substitua pela URL do seu MongoDB
7  const MONGO_HOST = 'mongodb://mongo:12qwaszx@localhost:27017/?authSource=mongodb';
8
9  const MONGO_DATABASE = 'mongodb';
10
11 const COLLECTION = "Cadastro";
12
13 async function connect() {
14     if (singleton) return singleton;
15
16     const client = new MongoClient(MONGO_HOST);
17     await client.connect();
18
19     singleton = client.db(MONGO_DATABASE);
20     return singleton;
21 }
22
23 //busca todos os registros da collection Cadastro
24 async function findAll() {
25     const db = await connect();
26     return db.collection(COLLECTION).find().toArray();
27 }
28
29 //busca registro por nome
30 async function findByName(nome) {
31     const query = { nome: nome };
32     const db = await connect();
33     return db.collection(COLLECTION).find(query).toArray();
34 }
35
36 //insere o registro
37 async function insertOne(query) {
38     const db = await connect();
39     return db.collection(COLLECTION).insertOne(query);
40 }
41
42 async function update(id, query) {
43     const db = await connect();
44     return db.collection(COLLECTION).updateOne({ _id: ObjectId.createFromHexString(id) }, { $set: query });
45 }
46
47 // permiti visualizar a funcao em outros arquivos
48 module.exports = [ findAll, findByName, insertOne, update ]
```

Buscar dados banco

► Editar arquivo server.js



```
server.js > ...
1 // Importa o módulo express
2 const express = require('express');
3
4 // Importa o módulo db
5 const db = require('./db');
6
7 // Cria uma aplicação express
8 const app = express();
9
10 // Middleware para analisar o JSON do corpo da requisição
11 app.use(express.json());
12
13 // Define a porta onde o servidor irá escutar
14 const port = 3000;
15
16 //rota que busca todos os usuarios
17 app.get('/usuarios', async (req, res, next) => {
18   try {
19     const docs = await db.findAll();
20     res.send(docs);
21   } catch (err) {
22     next(err);
23   }
24 });
25
26
27 //rota busca por nome de usuario
28 app.get('/buscarnome/:nome', async (req, res, next) => {
29   const nome = req.params.nome;
30   try {
31     const docs = await db.findByName(nome);
32     res.send(docs);
33   } catch (err) {
34     next(err);
35   }
36 });
37
38 //rota editar registro
39 app.get('/editar/:id', async (req, res, next) => {
40   const id = req.params.id
41   const { nome, idade, cidade } = req.body;
42   const query = { nome: nome ,idade: idade, cidade: cidade };
43
44   try {
45     const docs = await db.update(id, query);
46     res.send(docs);
47   } catch (err) {
48     next(err);
49   }
50 }
```


Download Postman

Download Postman

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

The Postman app

Download the app to get started with the Postman API Platform.

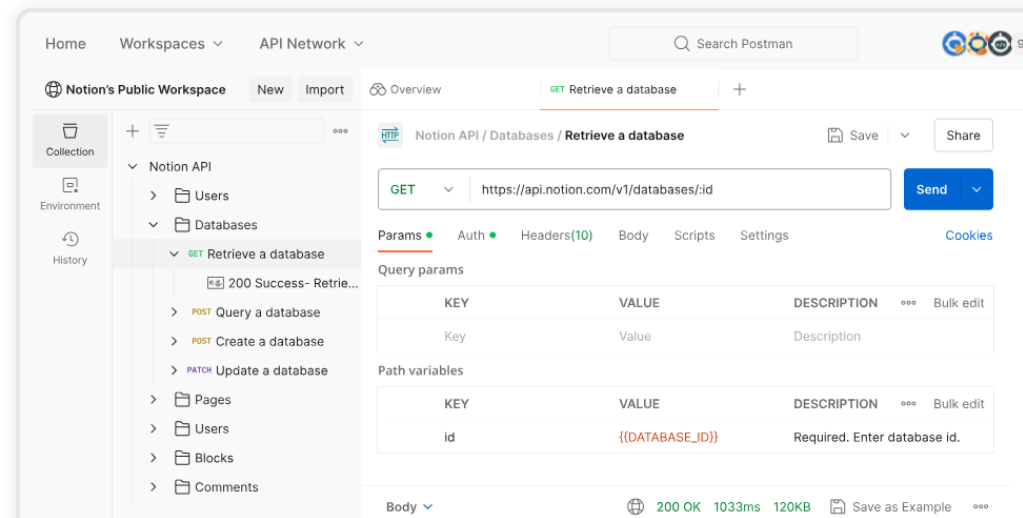
 **Windows 64-bit**

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

[Release Notes](#) →

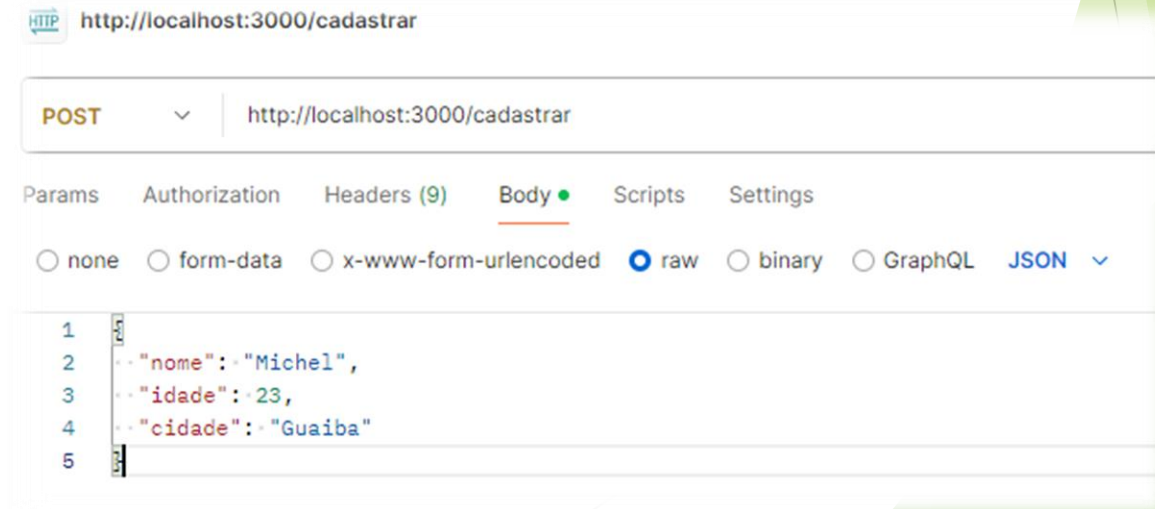
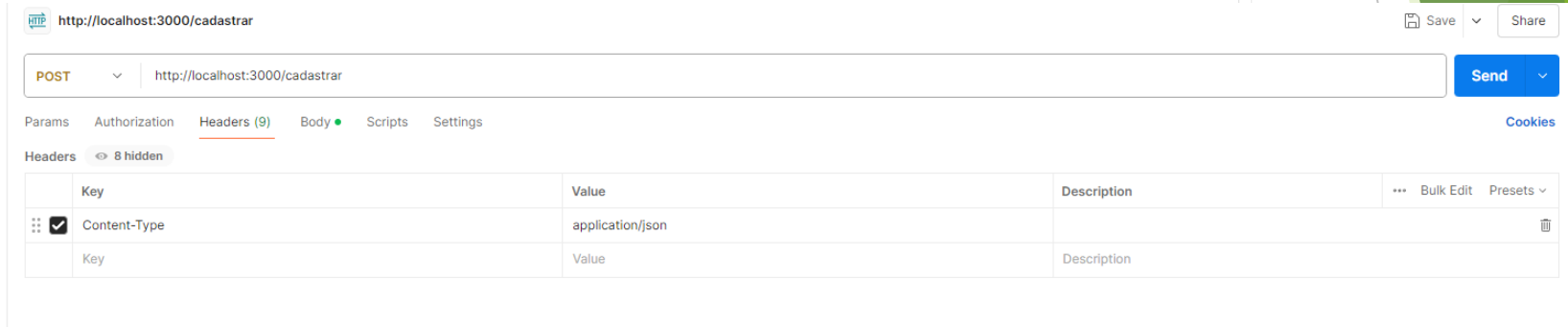
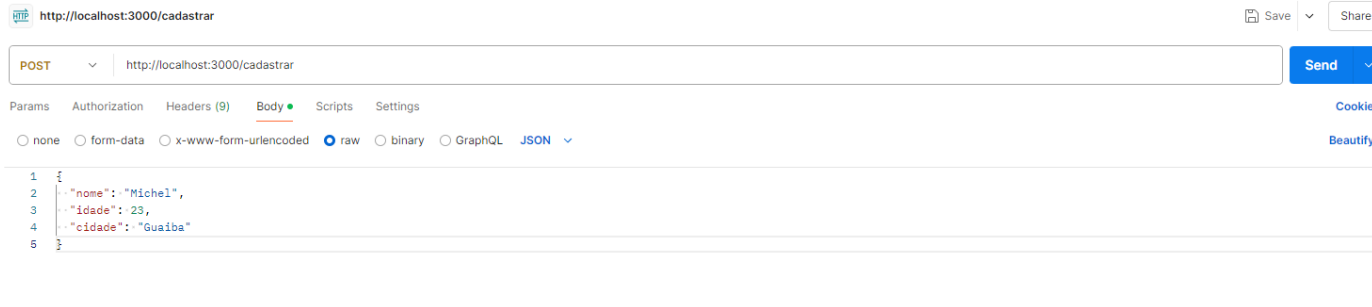
Not your OS? Download for Mac ([Intel Chip](#), [Apple Chip](#)) or Linux ([x64](#), [arm64](#))

Postman on the web



Postman

► Cadastrar registro



Postman

► Editor registro

http://localhost:3000/editar/66950cb47175daff5f49e1a9

GET http://localhost:3000/editar/66950cb47175daff5f49e1a9

Params Authorization Headers (9) **Body** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   ...
3   "nome": "Michel da Silva",
4   "idade": 30,
5   "cidade": "Guaiba"
6 }
```

http://localhost:3000/cadastrar

POST http://localhost:3000/cadastrar

Params Authorization **Headers (9)** Body Scripts Settings

Headers 8 hidden

	Key	Value	Description	...	Bulk Edit	Presets
...	<input checked="" type="checkbox"/> Content-Type	application/json				
	Key	Value	Description			

http://localhost:3000/editar/66950cb47175daff5f49e1a9

GET http://localhost:3000/editar/66950cb47175daff5f49e1a9

Params Authorization Headers (9) **Body** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   ...
3   "nome": "Michel da Silva",
4   "idade": 30,
5   "cidade": "Guaiba"
6 }
```

Revisão

- ▶ [Nodejs_Mongo.pdf](#)



Gamification

- ▶ [Nodejs_Mongo.pdf](#)