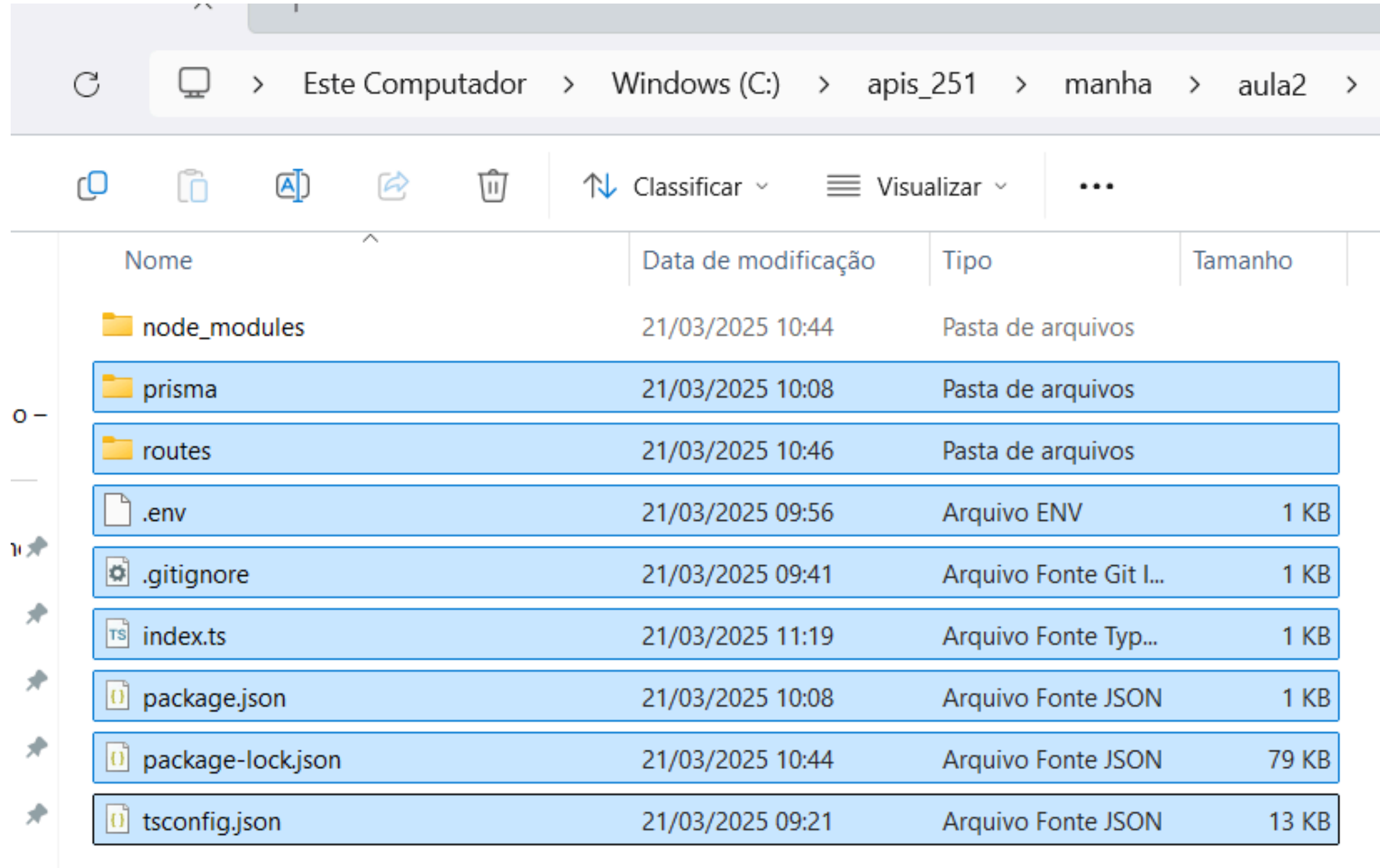




Desenvolvimento de **Serviços e APIs**

**Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas
Prof. Edécio Fernando Iepsen**

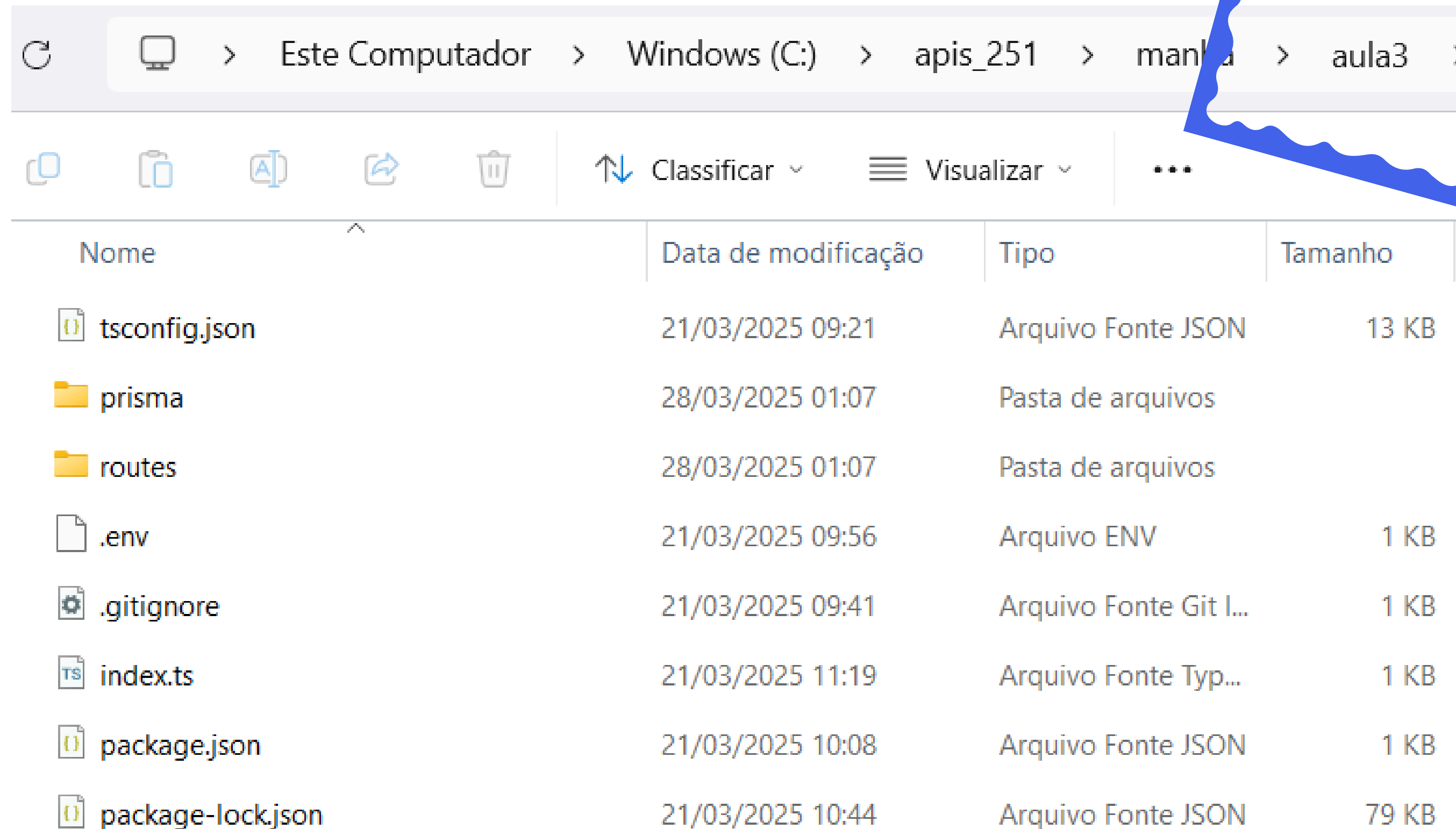
Novo projeto a partir de um existente











Nome	Data de modificação	Tipo	Tamanho
node_modules	21/03/2025 10:44	Pasta de arquivos	
prisma	21/03/2025 10:08	Pasta de arquivos	
routes	21/03/2025 10:46	Pasta de arquivos	
.env	21/03/2025 09:56	Arquivo ENV	1 KB
.gitignore	21/03/2025 09:41	Arquivo Fonte Git I...	1 KB
index.ts	21/03/2025 11:19	Arquivo Fonte Typ...	1 KB
package.json	21/03/2025 10:08	Arquivo Fonte JSON	1 KB
package-lock.json	21/03/2025 10:44	Arquivo Fonte JSON	79 KB
tsconfig.json	21/03/2025 09:21	Arquivo Fonte JSON	13 KB

Copie os arquivos básicos do projeto existente

Crie uma nova pasta e cole os arquivos



Nome	Data de modificação	Tipo	Tamanho
 tsconfig.json	21/03/2025 09:21	Arquivo Fonte JSON	13 KB
 prisma	28/03/2025 01:07	Pasta de arquivos	
 routes	28/03/2025 01:07	Pasta de arquivos	
 .env	21/03/2025 09:56	Arquivo ENV	1 KB
 .gitignore	21/03/2025 09:41	Arquivo Fonte Git I...	1 KB
 index.ts	21/03/2025 11:19	Arquivo Fonte Typ...	1 KB
 package.json	21/03/2025 10:08	Arquivo Fonte JSON	1 KB
 package-lock.json	21/03/2025 10:44	Arquivo Fonte JSON	79 KB

Vá ao prompt de comandos e rode o npm i

A screenshot of a Windows command prompt window. The text 'C:\apis_251\manha\aula3>npm i' is displayed in white on a black background. A blue, irregular, stamp-like graphic is overlaid on the right side of the terminal window, partially obscuring the text.

```
C:\apis_251\manha\aula3>npm i
```

Agora edite os arquivos e faça as alterações

Conecte-se ao MySQL e crie o banco de dados do projeto

```
C:\MariaDB107\bin>mysql -u root -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 210
Server version: 10.7.3-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database primevideo_manha;
Query OK, 1 row affected (0.014 sec)

MariaDB [(none)]> █
```

Ajuste o arquivo .env para o novo banco de dados

```
C: > apis_251 > manha > aula3 > ⚙ .env
```

```
1  DATABASE_URL="mysql://root:1234@localhost:3306/loja_filmes"
```

```
2
```


No schema.prisma, crie a model representando a tabela do banco de dados

```
9
10  model Filme {
11      id          Int          @id @default(autoincrement())
12      titulo      String       @db.VarChar(40)
13      genero      String       @db.VarChar(20)
14      duracao     Int          @db.SmallInt
15      preco       Decimal      @db.Decimal(7, 2)
16      datacad     DateTime     @default(now())
17      sinopse     String?
18      @@map("filmes")
19  }
```

Use campos enum para definir valores válidos

```
16  model Carro {
21      ano          Int          @db.SmallInt
22      preco         Decimal      @db.Decimal(10, 2)
23      combustivel   Combustivel  @default(FLEX)
24      km            Int          @db.MediumInt @default(0)
25      @@map("carros")
26  }
27
28  enum Combustivel {
29      FLEX
30      GASOLINA
31      ETANOL
32      DIESEL
33      ELETRICO
34  }
```


Rode o comando migrate para criar / alterar a tabela no banco

```
C:\apis_251\manha\aula2>npx prisma migrate dev --name "Criação Tabela Filmes"
Environment variables loaded from .env
Prisma schema loaded from prisma\schema.prisma
Datasource "db": MySQL database "loja_filmes" at "localhost:3306"
```

... e o generate para gerar o Prisma Client

```
C:\apis_251\manha\aula2>npx prisma generate
Environment variables loaded from .env
Prisma schema loaded from prisma\schema.prisma

✓Generated Prisma Client (v6.5.0) to .\node_modules\@prisma\client in 62ms

Start by importing your Prisma Client (See: https://pris.ly/d/importing-client)
```

Crie os métodos/rotas para realizar o CRUD

```
C: > apis_251 > noite > aula2 > routes > TS filmes.ts > ...  
1  import { PrismaClient } from "@prisma/client"  
2  import { Router } from "express"  
3  
4  const prisma = new PrismaClient()  
5  const router = Router()  
6  
7  router.get("/", async (req, res) => {  
8      try {  
9          const filmes = await prisma.filme.findMany({  
10             orderBy: { id: 'desc' },  
11             })  
12             res.status(200).json(filmes)  
13         } catch (error) {  
14             res.status(500).json({ erro: error })  
15         }  
16     })
```

```
19 router.post("/", async (req, res) => {
20     // recebe as variáveis vindas no corpo da requisição
21     const { titulo, genero, duracao, preco, sinopse = null } = req.body
22
23     // verifica se campos obrigatórios foram passados
24     if (!titulo || !genero || !duracao || !preco) {
25         res.status(400).json({ erro: "Informe todos os dados" })
26         return
27     }
28
29     try {
30         const filme = await prisma.filme.create({
31             data: { titulo, genero, duracao, preco, sinopse }
32         })
33         res.status(201).json(filme)
34     } catch (error) {
35         res.status(400).json({ erro: error })
36     }
37 })
```

```
39
40 router.put("/:id", async (req, res) => {
41     // recebe o id passado como parâmetro
42     const { id } = req.params
43
44     // recebe as variáveis vindas no corpo da requisição
45     const { titulo, genero, duracao, preco, sinopse } = req.body
46
47     // verifica se os campos obrigatórios foram passados
48     if (!titulo || !genero || !duracao || !preco) {
49         res.status(400).json({ erro: "Informe todos os dados" })
50         return
51     }
52
53     try {
54         const filme = await prisma.filme.update({
55             where: { id: Number(id) },
56             data: { titulo, genero, duracao, preco, sinopse }
57         })
58         res.status(200).json(filme)
59     } catch (error) {
60         res.status(400).json({ erro: error })
61     }
62 })
```

```
64 router.delete("/:id", async (req, res) => {
65     // recebe o id passado como parâmetro
66     const { id } = req.params
67
68     // realiza a exclusão do filme
69     try {
70         const filme = await prisma.filme.delete({
71             where: { id: Number(id) }
72         })
73         res.status(200).json(filme)
74     } catch (error) {
75         res.status(400).json({ erro: error })
76     }
77 })
78
79 export default router
```

```
C:\apis_251\manha\aula2>npm run dev
```

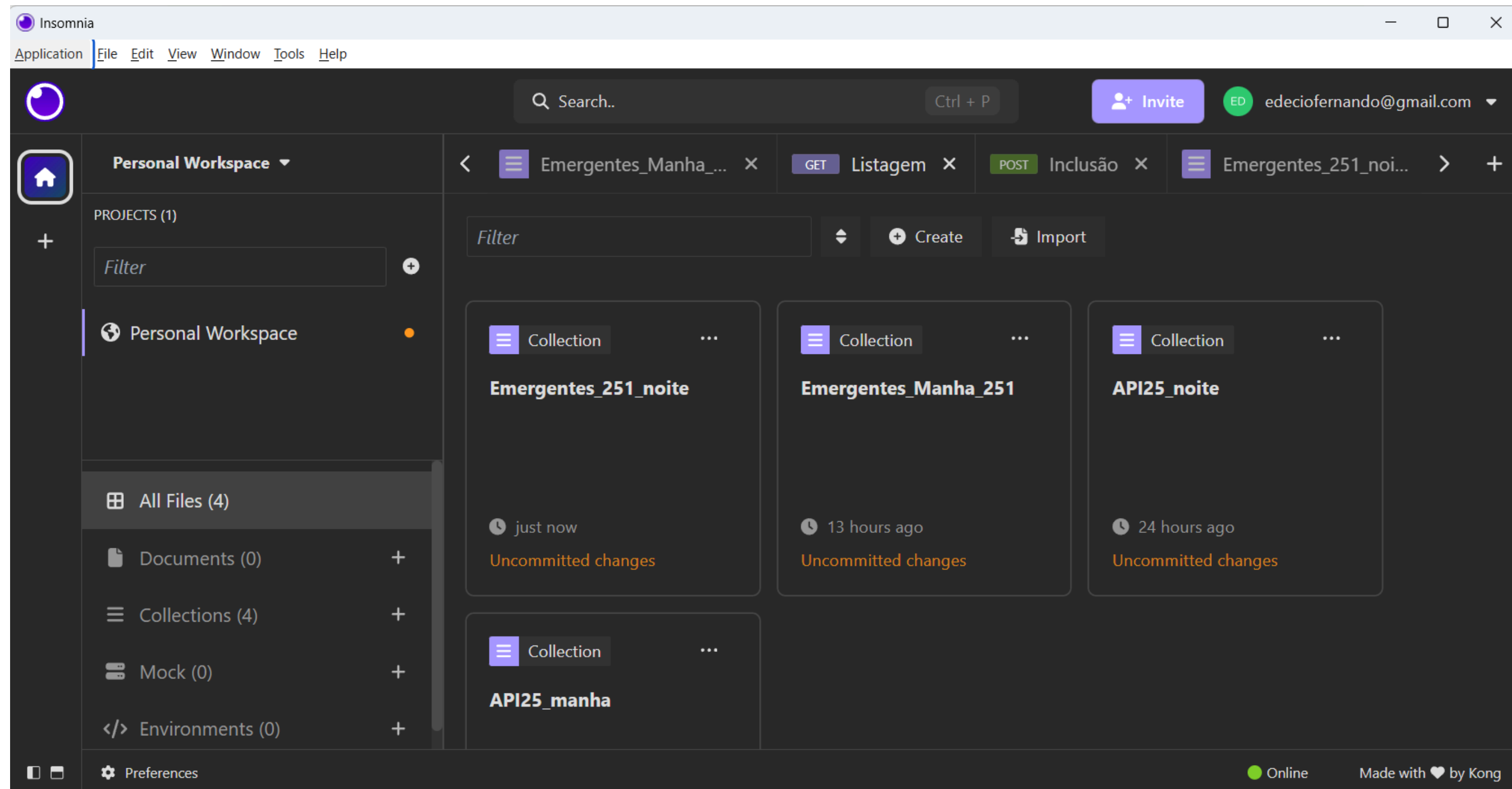
```
> aula2@1.0.0 dev
```

```
> npx ts-node-dev --respawn index.ts
```

```
[INFO] 09:01:36 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2,  
typescript ver. 5.8.2)
```

```
Servidor rodando na porta: 3000
```


Insomnia



Ferramenta para testes e desenvolvimento de APIs REST, GraphQL e outras.

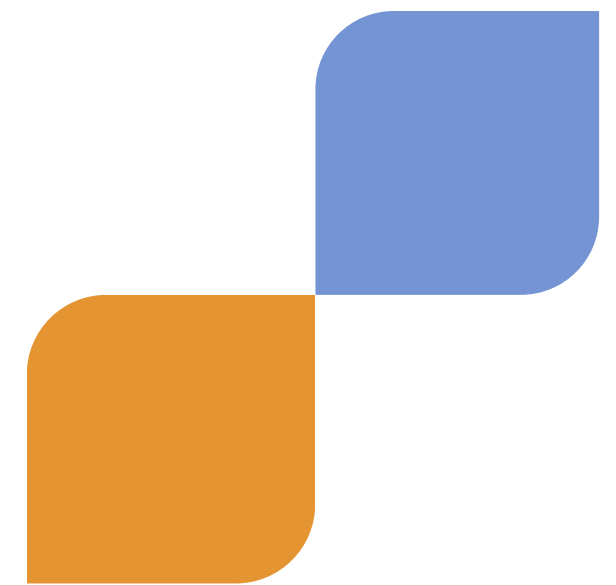
O que é o Insomnia?

- **Insomnia é um software para testes de APIs.**
- **Permite enviar requisições HTTP e visualizar respostas de APIs.**
- **Suporta REST, GraphQL, WebSockets e outros protocolos.**
- **Disponível para Windows, macOS e Linux.**



Principais Funcionalidades

- **Envio de Requisições:** Suporte a GET, POST, PUT, DELETE e outros.
- **Autenticação:** Tokens, OAuth, Basic Auth e API Keys.
- **Variáveis e Ambiente:** Facilita testes em diferentes cenários.
- **Histórico de Requisições:** Facilita revisão e reutilização.



Inclusão: POST

The screenshot displays the Insomnia REST client interface. The top bar shows the application name 'Insomnia' and a menu with 'Application', 'File', 'Edit', 'View', 'Window', 'Tools', and 'Help'. A search bar and a user profile 'edeciofernando@gmail.com' are also visible.

The main workspace is divided into several panels:

- Left Panel:** Contains a sidebar with a home icon, a plus sign, and a list of environments: 'Base Environment', 'Add Cookies', and 'Add Certificates'. Below this is a 'Filter' input and a folder named 'Aula 2 - Filmes'. Inside this folder, there are four API endpoints: 'Alteração' (PUT), 'Exclusão' (DEL), 'Inclusão' (POST), and 'Listagem' (GET). At the bottom of the sidebar is a 'master' environment icon and a 'Preferences' button.
- Top Panel:** Shows the current API endpoint 'localhost:3000/filmes' with a 'POST' method. A 'Send' button is present. To the right, a status bar indicates '201 Created', '185 ms', and '280 B'.
- Body Panel:** The 'Body' tab is selected, showing a JSON payload:

```
1 {
2   "titulo": "Ainda Estou Aqui",
3   "genero": "Drama",
4   "duracao": 136,
5   "preco": 19.9,
6   "sinopse": "Ambientado na ditadura militar brasileira, o filme aborda a história real do desaparecimento e morte do engenheiro civil e ex-deputado Rubens Paiva"
7 }
```
- Preview Panel:** The 'Preview' tab is selected, showing the response JSON:

```
1 {
2   "id": 1,
3   "titulo": "Ainda Estou Aqui",
4   "genero": "Drama",
5   "duracao": 136,
6   "preco": "19.9",
7   "datacad": "2025-03-28T13:03:03.621Z",
8   "sinopse": "Ambientado na ditadura militar brasileira, o filme aborda a história real do desaparecimento e morte do engenheiro civil e ex-deputado Rubens Paiva"
9 }
```

At the bottom of the interface, there is a status bar with 'Online' and 'Made with ❤ by Kong'.

Listagem: GET

The screenshot displays the Insomnia REST client interface. The top bar includes the Insomnia logo, a search bar, a keyboard shortcut (Ctrl + P), an 'Invite' button, and a user profile (edeciofernando@gmail.com). The main workspace is divided into several sections:

- Left Sidebar:** Contains a home icon, a plus sign, and a list of environments and collections. The 'API25_manha' collection is selected, showing a list of endpoints: 'Alteração' (PUT), 'Exclusão' (DEL), 'Inclusão' (POST), and 'Listagem' (GET). The 'Listagem' endpoint is highlighted.
- Top Bar:** Shows the current request details: 'GET localhost:3000/filmes'. The 'Send' button is visible.
- Response Tab:** The 'Preview' tab is active, showing the response status '200 OK' with a response time of '18 ms' and a size of '563 B'. The response body is a JSON array of movie objects.
- Response Body:** The JSON response is displayed in a syntax-highlighted format:

```
1 [
2   {
3     "id": 1,
4     "titulo": "Ainda Estou Aqui",
5     "genero": "Drama",
6     "duracao": 136,
7     "preco": "19.9",
8     "datacad": "2025-03-28T13:03:03.621Z",
9     "sinopse": "Ambientado na ditadura militar brasileira,
10    o filme aborda a história real do desaparecimento e morte
11    do engenheiro civil e ex-deputado Rubens Paiva"
12  },
13  {
14    "id": 2,
15    "titulo": "A Era do Gelo",
16    "genero": "Animação",
17  }
18 ]
```

Exclusão: DELETE

The screenshot displays the Insomnia REST client interface. The top bar shows the application name 'Insomnia' and a search bar. The left sidebar contains a list of collections, including 'API25_manha'. The main panel shows a DELETE request to 'localhost:3000/filmes/3'. The response is a 200 OK status with a 50 ms duration and 136 B of data. The response body is a JSON object:

```
1 {
2   "id": 3,
3   "titulo": "A Era do Gelo 2",
4   "genero": "Animação",
5   "duracao": 90,
6   "preco": "6.9",
7   "datacad": "2025-03-28T12:06:00.743Z",
8   "sinopse": null
9 }
```

The interface also includes a 'Send' button, a 'Preview' tab, and a 'Mock' button. The bottom status bar shows the current collection 'master' and a JSON path '\$.store.books[*].author'.

Alteração: PUT

The screenshot displays the Insomnia REST client interface. The top bar shows the application name 'Insomnia' and a search bar. The left sidebar contains a navigation menu with a home icon, a plus sign, and a list of collections: 'API25_manha', 'Aula 2 - Filmes', and 'APIs25_manha'. The 'API25_manha' collection is selected, showing a list of endpoints: 'Alteração' (PUT), 'Exclusão' (DEL), 'Inclusão' (POST), and 'Listagem' (GET). The main panel shows a PUT request to 'localhost:3000/filmes/2'. The request body is a JSON object with the following fields: 'titulo', 'genero', 'duracao', 'preco', and 'sinopse'. The response is a 200 OK status with a response time of 532 ms and a body size of 156 B. The response body is a JSON object with the same fields as the request, plus an 'id' field and a 'datacad' field. The bottom bar shows the 'Beautify JSON' button and the JSON path '\$.store.books[*].author'.

Insomnia

Application File Edit View Window Tools Help

Search.. Ctrl + P Invite edeciofernando@gmail.com

< API25_manha Run

Base Environment Add Cookies Add Certificates

Filter

Aula 2 - Filmes

PUT Alteração DEL Exclusão POST Inclusão GET Listagem

APIs25_manha

master

Beautify JSON

PUT localhost:3000/filmes/2 Send

200 OK 532 ms 156 B Just Now

Params Body Auth Headers 4 Scripts Docs

JSON

```
1 {
2   "titulo": "A Era do Gelo",
3   "genero": "Animação",
4   "duracao": 93,
5   "preco": 9.90,
6   "sinopse": "Vinte mil anos atrás..."
7 }
```

Preview

```
1 {
2   "id": 2,
3   "titulo": "A Era do Gelo",
4   "genero": "Animação",
5   "duracao": 93,
6   "preco": "9.9",
7   "datacad": "2025-03-28T12:04:49.635Z",
8   "sinopse": "Vinte mil anos atrás..."
9 }
```

Preview

\$.store.books[*].author