

SW-II

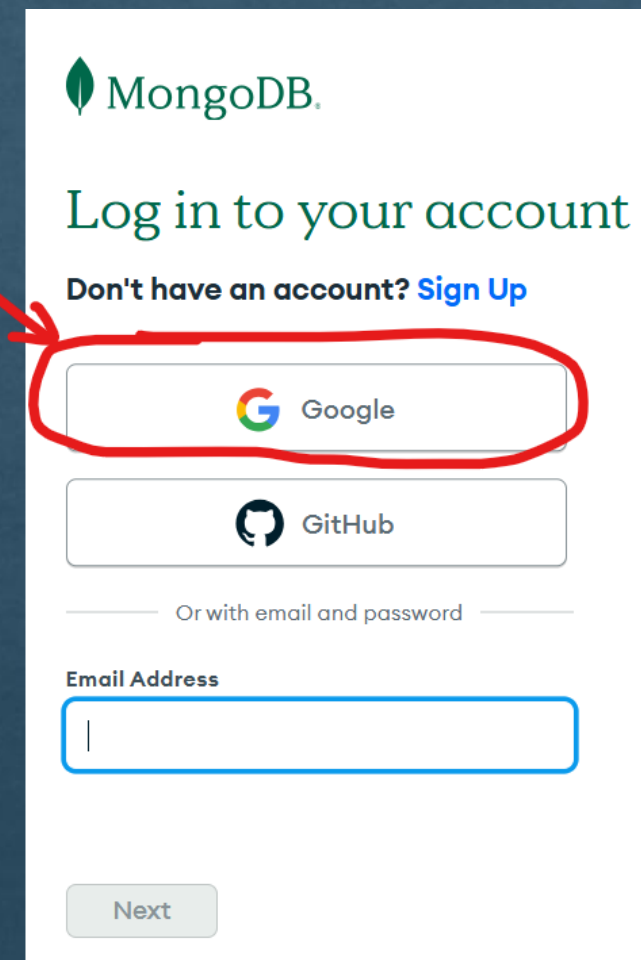
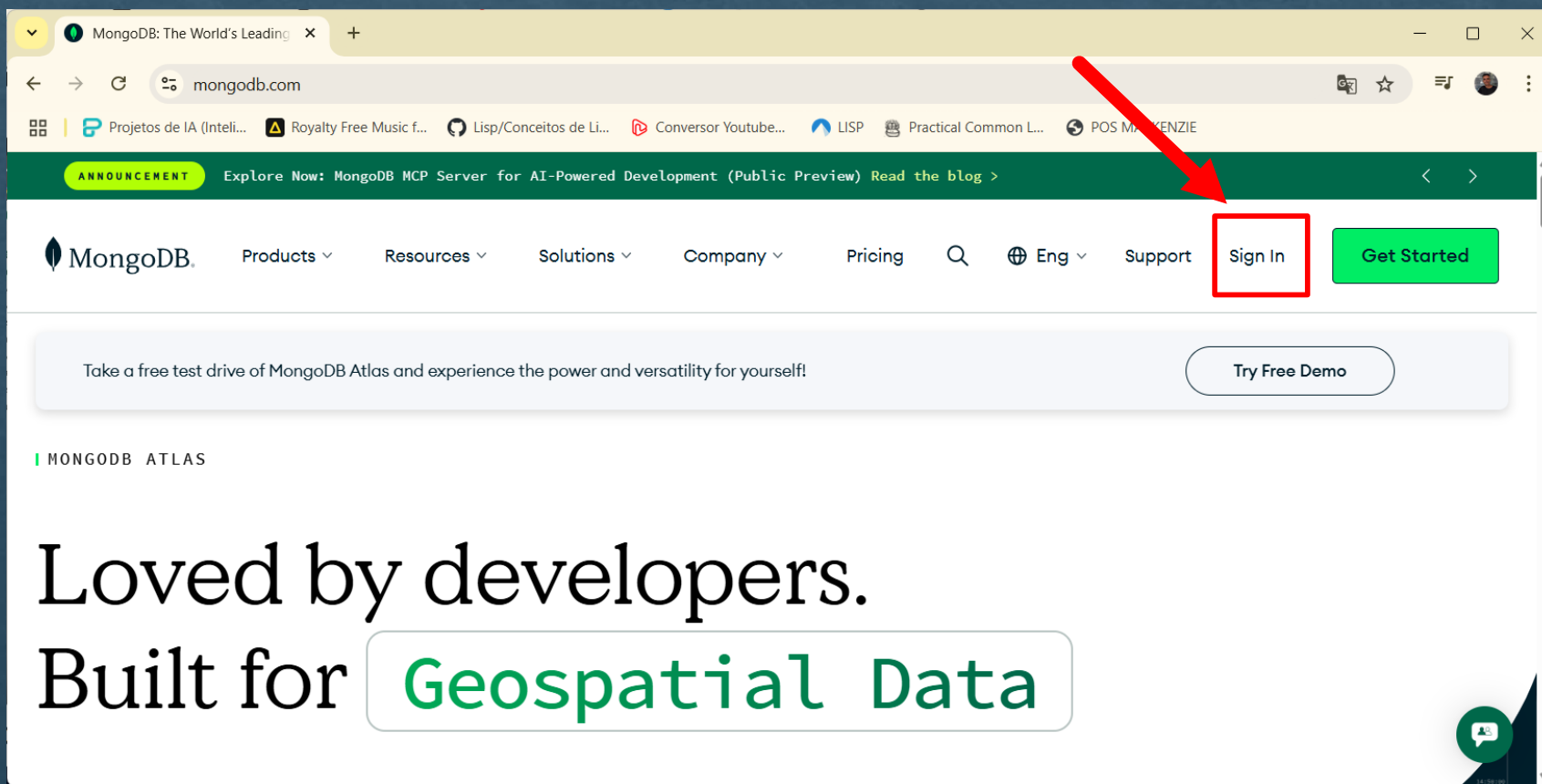
SISTEMAS WEB II

Prof. Anderson Vanin

AULA 14 – APIs com NODE e Banco de Dados

Criar uma conta no MongoDB

- <https://www.mongodb.com/>



Deploy your cluster

Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.



M10

\$0.12/hour

Dedicated cluster for development environments and low-traffic applications.

STORAGE

10 GB

RAM

2 GB

vCPU

2 vCPUs



Flex

From \$0.011/hour

Up to \$30/month

For development and testing, with on-demand burst capacity for unpredictable traffic.

STORAGE

5 GB

RAM

Shared

vCPU

Shared



Free

For learning and exploring MongoDB in a cloud environment.

STORAGE

512 MB

RAM

Shared

vCPU

Shared



Free forever! Your free cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Configurations

Name

You cannot change the name once the cluster is created.

PrimeiroCluster

Provider



Region



Sao Paulo (sa-east-1) ★

★ Recommended ⓘ

🌿 Low carbon emissions ⓘ

Tag (optional)

Create your first tag to categorize and label your resources; more tags can be added later. [Learn more.](#)

Select or enter key

:

Select or enter value

Quick setup

☒ Automate security setup ⓘ

☒ Preload sample dataset ⓘ

I'll do this later

Go to Advanced Configuration

Create Deployment

Connect to PrimeiroCluster ✕



You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

1. Add a connection IP address

✓ Your current IP address (179.212.46.44) has been added to enable local connectivity. Only an IP address you add to your Access List will be able to connect to your project's clusters. Add more later in [Network Access](#).

2. Create a database user

This first user will have [atlasAdmin](#) permissions for this project.

We autogenerated a username and password. You can use this or create your own.

i You'll need your database user's credentials in the next step. Copy the database user password.

Username

etecmcm2025

Password

[Redacted password]

HIDE

Copy

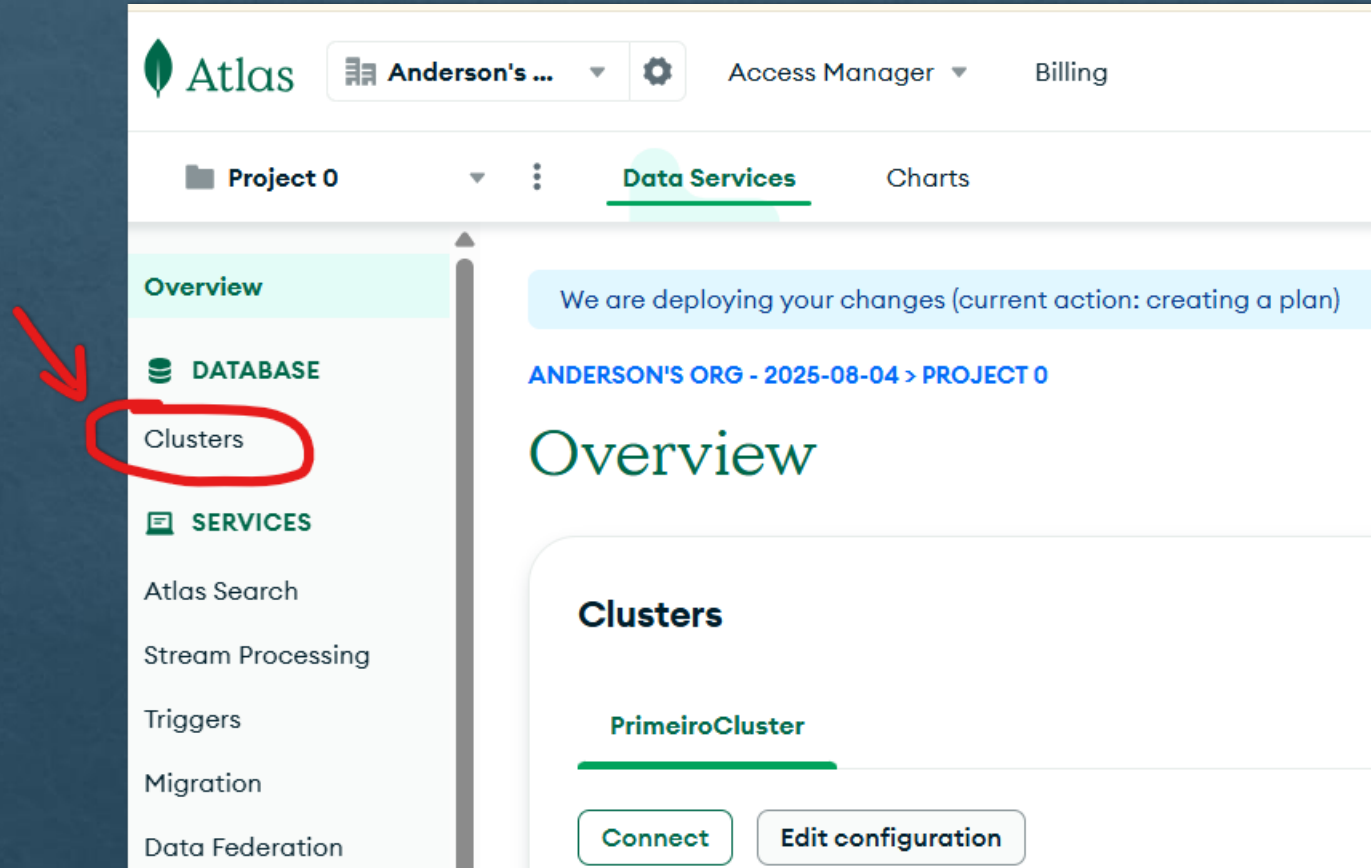
Create Database User

Suas credenciais

Importante anotar:

- Nome de Usuário do Banco de Dados
- Senha do usuário do Banco de Dados
- String de Conexão.

String de Conexão



String de Conexão

Clusters

Find a database deployment...

Edit Config

+ C

✓ Sample dataset successfully loaded. Access it in [Data Explorer](#) or by connecting with the MongoDB Shell.

● PrimeiroCluster

Connect

View Monitoring

Browse Collections

...

Visualize Your Data

Build dashboards and charts, and embed them in your apps with MongoDB Charts.

Explore Charts

Dismiss

● R 0

● W

Last 2 minutes

100.0/s

i

Connections 3.0

Last 2 minutes

3.0

i

● In 301.32 KB/s

● Out 2.08 KB/s

Last 2 minutes

301.32 KB/s

i

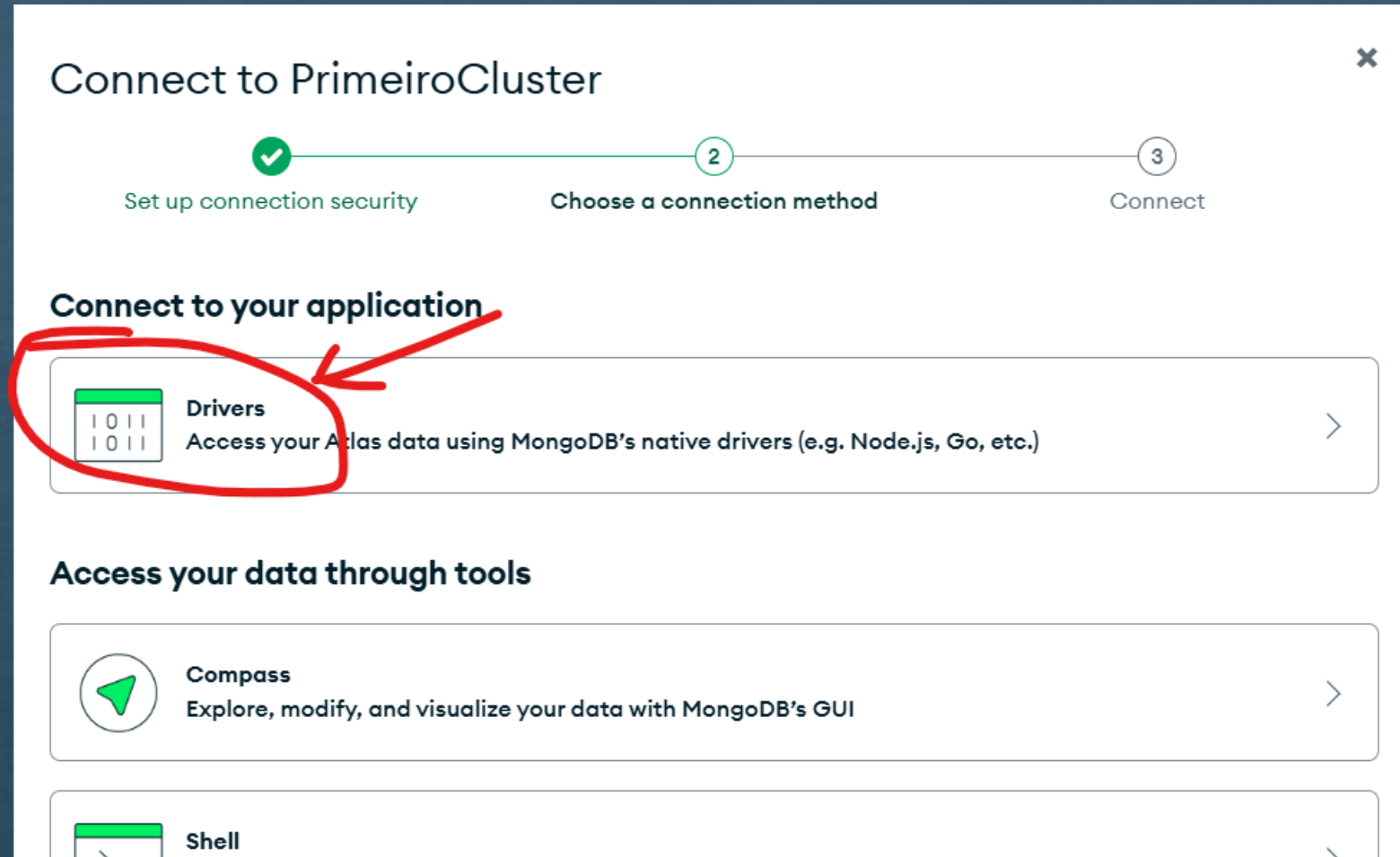
Data Size

28.12 MB / 512.00 MB (5%)

Last 2 minutes

512.00 MB

String de Conexão



String de Conexão

×

✓

✓

3

Set up connection securityChoose a connection methodConnect

Connect to PrimeiroCluster

Connecting with MongoDB Driver

1. Select your driver and version

We recommend installing and using the latest driver version.

Driver	Version
Node.js	6.7 or later

2. Install your driver

Run the following on the command line

```
npm install mongodb
```

[View MongoDB Node.js Driver installation instructions.](#)

3. Add your connection string into your application code

Use this connection string in your application

☐ View full code sample

```
mongodb+srv://etecmcm2025:<db_password>@primeirocluster.yptzrns.mongodb.net/?retryWrites=true&w=majority&appName=PrimeiroCluster
```

Suas credenciais

Importante anotar:

- user: etecmcm2025
- password: xxxxxxxx
- string:
`mongodb+srv://etecmcm2025:<db_password>@primeirocluster.yptzrns.mongodb.net/?retryWrites=true&w=majority&appName=PrimeiroCluster`

Habilitando o acesso de qualquer IP

SERVICES

Atlas Search

Stream Processing

Triggers

Migration

Data Federation

SECURITY

Quickstart

Backup

Database Access

Network Access

Advanced

New On Atlas 3

Network Access

IP Access List

Peering

Private Endpoint

+ ADD IP ADDRESS

! You will only be able to connect to your cluster from the following list of IP Addresses:

IP Address	Comment	Status	Actions
179.212.46.44/32 (includes your current IP address)	Created as part of the Auto Setup process	● Active	EDIT DELETE

Habilitando o acesso de qualquer IP

Edit IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more](#)

ALLOW ACCESS FROM ANYWHERE

Access List Entry:

179.212.46.44/32

Comment:

Created as part of the Auto Setup process

Cancel

Confirm

Edit IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more](#)

ALLOW ACCESS FROM ANYWHERE

Access List Entry:

0.0.0.0/0

Comment:

Created as part of the Auto Setup process

Cancel

Confirm

Instalar a biblioteca Prisma

- `npm install prisma --save-dev`

```
C:\Users\Anderson\Desktop\api_node>npm install prisma --save-dev
```

```
added 53 packages, and audited 121 packages in 13s
```

```
26 packages are looking for funding  
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
{ } package.json > ...  
18 |   | https://github.com/ProfA  
19 |   },  
20 |   "homepage": "https://github.com/ProfA",  
21 |   "dependencies": {  
22 |     | "express": "^5.1.0"  
23 |   },  
24 |   "devDependencies": {  
25 |     | "prisma": "^6.13.0"   
26 |   }  
27 | }
```


Instalar a biblioteca Prisma

- `npx prisma init`

```
C:\Users\Anderson\Desktop\api_node>npx prisma init
```

```
Fetching latest updates for this subcommand...
```

```
✓ Your Prisma schema was created at prisma/schema.prisma  
  You can now open it in your favorite editor.
```

```
warn You already have a .gitignore file. Don't forget to add `.env`  
ion
```

▼ API_NODE

> node_modules

> prisma

⚙ .env

📄 .gitignore

{ } package-lock.json

{ } package.json

📖 README.md

JS server.js

Editar o arquivo .env

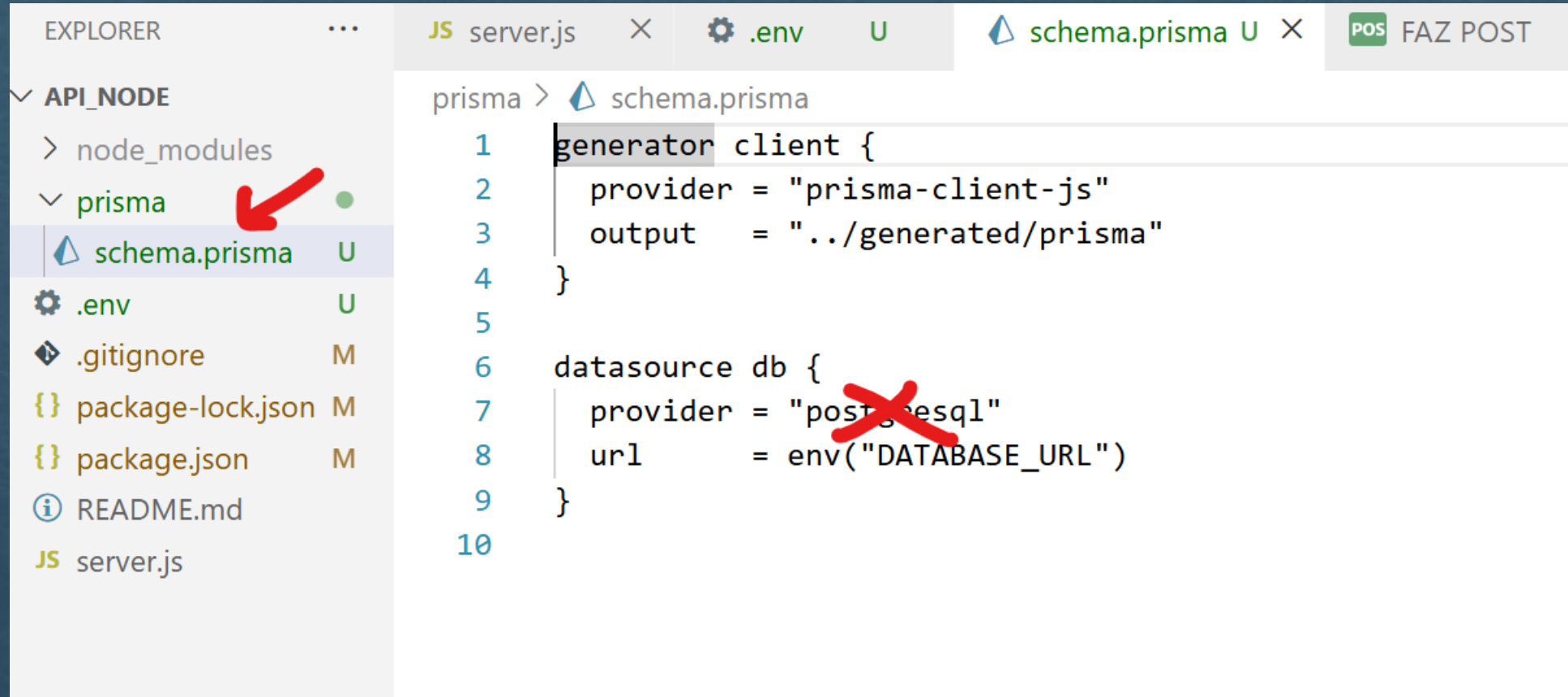
- Neste arquivo vamos inserir a nossa url de conexão com o mongodb



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the file structure of a project named 'API_NODE'. The files listed are 'node_modules', 'prisma', '.env' (highlighted with a gear icon and a 'U' status), '.gitignore', 'package-lock.json', 'package.json', 'README.md', and 'server.js'. The main editor area shows the '.env' file with a single line of code: `1 DATABASE_URL="mongodb+srv://andersonvanin:[REDACTED]@clusterapinode.npaa5vz.mongodb.net/ClusterApiNode?retryWrites=true&w=majority&appName=ClusterApiNode"`. The top of the editor shows several tabs: 'server.js', '.env', 'FAZ POST', and 'FAZ GET'. The '.env' tab is active, and the code is highlighted in red.

```
1 DATABASE_URL="mongodb+srv://andersonvanin:[REDACTED]@clusterapinode.npaa5vz.mongodb.net/ClusterApiNode?retryWrites=true&w=majority&appName=ClusterApiNode"
```

Alterar o Schema (Estrutura de seu banco)



The screenshot shows the Visual Studio Code interface with the Explorer on the left and the Editor on the right. In the Explorer, the 'API_NODE' folder is expanded, showing 'node_modules', 'prisma', 'schema.prisma', '.env', '.gitignore', 'package-lock.json', 'package.json', 'README.md', and 'server.js'. A red arrow points to the 'prisma' folder. The 'schema.prisma' file is selected and its content is displayed in the Editor. The Prisma schema defines a client generator and a database data source. A red 'X' is drawn over the word 'postgres' in the 'datasource db' block, indicating a modification to the database provider.

```
prisma > schema.prisma
1 generator client {
2   provider = "prisma-client-js"
3   output   = "../generated/prisma"
4 }
5
6 datasource db {
7   provider = "postgres"
8   url      = env("DATABASE_URL")
9 }
10
```

Alterar o Schema (Estrutura de seu banco)

prisma >  schema.prisma

```
1  generator client {  
2    provider = "prisma-client-js"  
3    output   = "../generated/prisma"  
4  }  
5  
6  datasource db {  
7    provider = "mongodb"  
8    url      = env("DATABASE_URL")  
9  }  
10  
11  model Usuario {  
12    id      String  @id @default(auto()) @map("_id") @db.ObjectId  
13    email   String  @unique  
14    nome    String  
15    idade   String  
16  }
```




Salvar alterações e rodar comando de implantação

- `npx prisma db push`

```
C:\Users\Anderson\Desktop\api_node>npx prisma db push
Environment variables loaded from .env
Prisma schema loaded from prisma\schema.prisma
Datasource "db": MongoDB database "ClusterApiNode" at "clusterapin
Applying the following changes:

[+] Collection `Usuario`
```

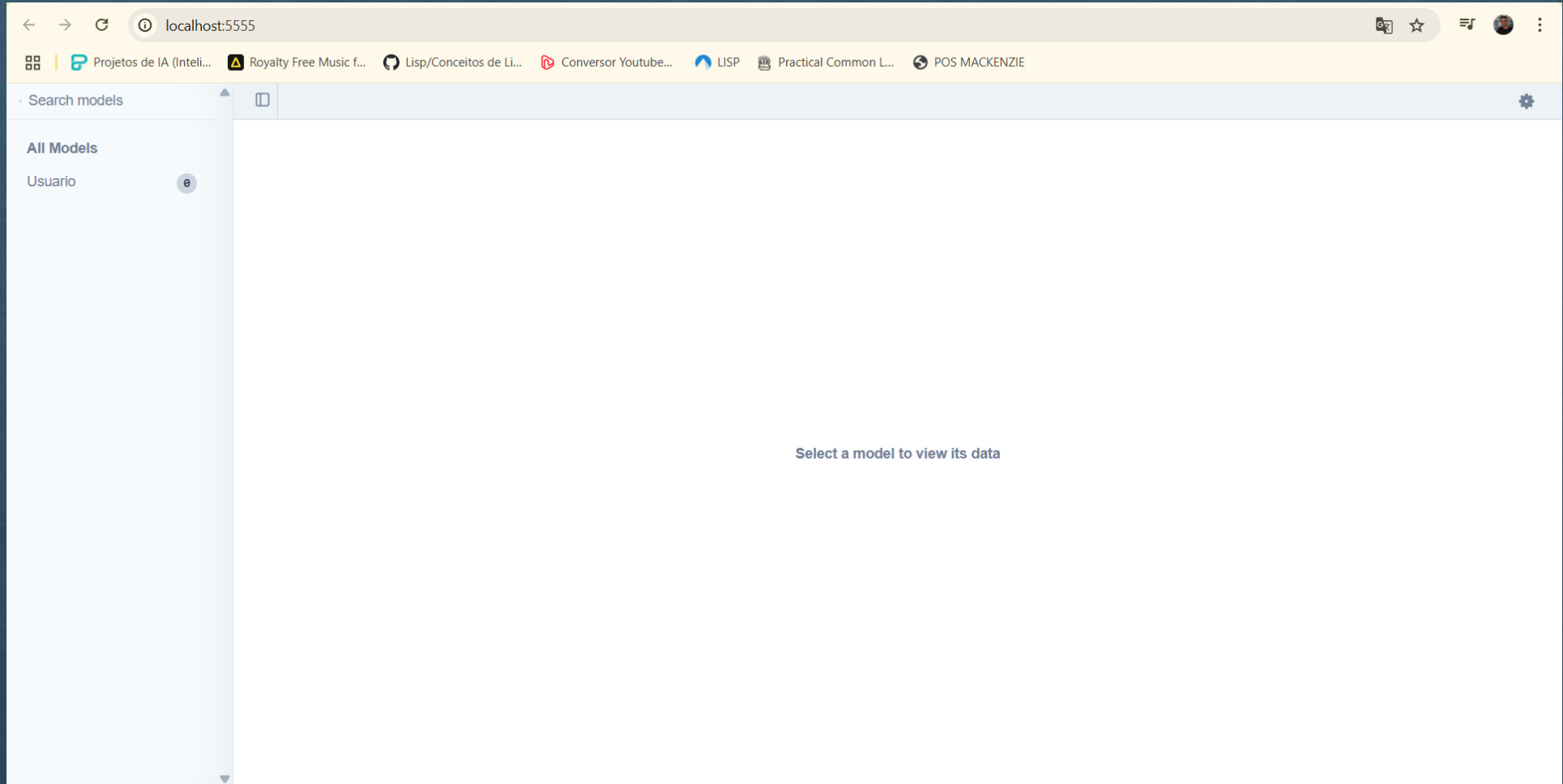


Instalar o Prisma Client

- `npm install @prisma/client`

Rodar o Prisma Client

- npx prisma studio



Rodar o Prisma Client

- Adicione um usuário

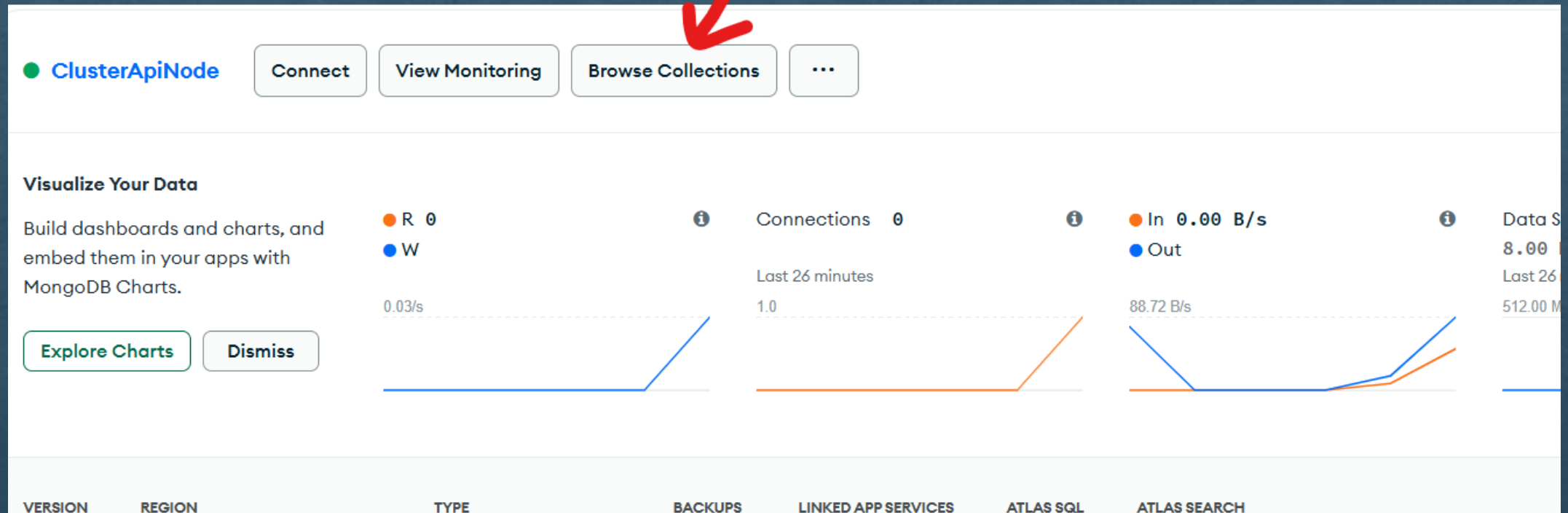
Search models

All Models

Usuario 1

Usuario X				
<div><div>Filters</div><div>None</div><div>Fields</div><div>All</div><div>Showing</div><div>1 of 1</div><div>Add record</div></div>				
<input type="checkbox"/>	id A	email A	nome A	idade A
<input type="checkbox"/>	6899ed759774a6b9ae8b0226	anderson@email.com	Anderson	51

Verificar a coleção criada no MongoDB



Verificar a coleção criada no MongoDB

The screenshot displays the MongoDB Atlas interface. The top navigation bar includes tabs for Overview, Real Time, Metrics, Collections (which is the active tab), Atlas Search, Query Insights, and Performance Advisor. Below the navigation bar, the status 'DATABASES: 1 COLLECTIONS: 1' is shown, along with a 'PREVIEW' button and a link to 'New Data Explorer'.

On the left sidebar, there is a '+ Create Database' button and a 'Search Namespaces' search bar. Under the 'ClusterApiNode' namespace, the 'Usuario' collection is highlighted with a red arrow.

The main content area shows the details for the 'ClusterApiNode.Usuario' collection. It includes statistics: 'STORAGE SIZE: 20KB', 'LOGICAL DATA SIZE: 85B', 'TOTAL DOCUMENTS: 1', and 'INDEXES TOTAL SIZE: 40KB'. Below these are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns' (with a count of 0), 'Aggregation', and 'Search Indexes'. A link to 'Generate queries from natural language in Compass' is also present.


A 'Filter' section allows users to 'Type a query: { field: 'value' }'. Below this, the 'QUERY RESULTS: 1-1 OF 1' section displays a single document. A red arrow points to the document's details:

```
{
  "_id": ObjectId('6899ed759774a6b9ae8b0226'),
  "email": "anderson@email.com",
  "nome": "Anderson",
  "idade": "51"
}
```

CRIANDO UM NOVO USUÁRIO VIA API E
SALVANDO NO BANCO DE DADOS MONGODB

Importar o Prisma Client no arquivo da api

```
JS server.js  ●  POS FAZ POST  GET FAZ GET
JS server.js > [🔗] usuarios
1  import express from 'express'
2
3  const { PrismaClient } = require('@prisma/client')
4  const prisma = new PrismaClient()
5
6
7
8
```




Criando a rota de POST

```
JS server.js > ...
1  import express from 'express'
2
3  const { PrismaClient } = require('@prisma/client')
4  const prisma = new PrismaClient()
5
6  const app = express()
7  app.use(express.json())
8
9  const usuarios = []
10
11  //Criando uma rota
12  app.post('/usuarios', async (req,res)=>{
13
14      await prisma.usuarios.create({
15          data:{
16              email:req.body.email,
17              nome:req.body.nome,
18              idade:req.body.idade
19          }
20      })
21
22      res.status(201).json(req.body)
23  })
24
```

Algumas alterações que podem ser necessárias

```
prisma > schema.prisma
1  generator client {
2    | provider = "prisma-client-js"
3  }
4
5
6  datasource db {
7    | provider = "mongodb"
8    | url      = env("DATABASE_URL")
9  }
10
11 model Usuario {
12   | id      String  @id @default(auto()) @map("_id") @db.ObjectId
13   | email   String  @unique
14   | nome    String
15   | idade   String
16 }
```



Algumas alterações que podem ser necessárias


```
JS server.js > ...  
1  import express from 'express'  
2  
3  import pkg from '@prisma/client';  
4  const { PrismaClient } = pkg;  
5  const prisma = new PrismaClient();  
6  
7  const app = express()  
8  app.use(express.json())  
9  
10 //Criando uma rota  
11 app.post('/usuarios', async (req,res)=>{  
12  
13     await prisma.usuarios.create({  
14         data:{  
15             email:req.body.email,
```



Após as alterações rodar o comando

- `npx prisma generate`

Testar com uma nova requisição POST



POST Send

Query Headers² Auth Body¹ Tests Pre Run

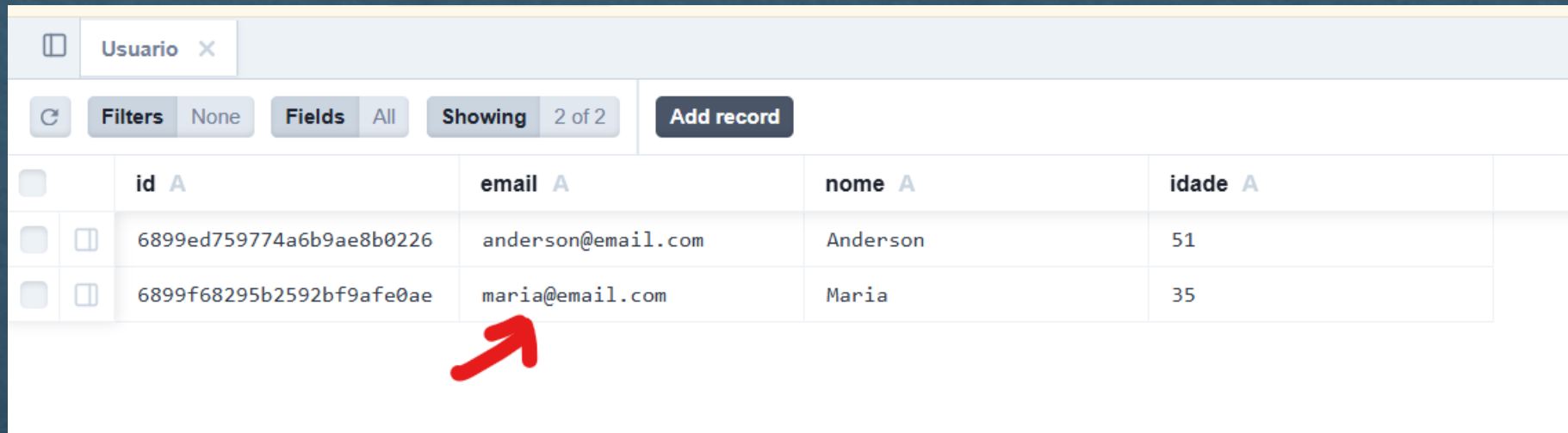
JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {  
2   "nome": "Maria",  
3   "idade": "35",  
4   "email": "maria@email.com"  
5 }
```


Verificar o usuário criado no Prisma Studio

- `npx prisma studio`



The screenshot shows the Prisma Studio interface for a database table named 'Usuario'. The table has four columns: 'id', 'email', 'nome', and 'idade'. There are two records displayed. A red arrow points to the 'email' field of the second record, 'maria@email.com'.

	id A	email A	nome A	idade A
<input type="checkbox"/>	6899ed759774a6b9ae8b0226	anderson@email.com	Anderson	51
<input type="checkbox"/>	6899f68295b2592bf9afe0ae	maria@email.com	Maria	35

LISTANDO TODOS OS USUÁRIO VIA API

Alterar a rota de GET

```
24 app.get('/usuarios', async (req, res)=>{
25
26     const usuarios_db = await prisma.usuario.findMany()
27
28     res.status(200).json(usuarios_db)
29 })
30
31 //Porta de resposta
32 app.listen(3000)
```

Testar a requisição de GET

GET ▼ localhost:3000/usuarios Send

Query

Headers ²

Auth

Body

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 200 OK Size: 183 Bytes Time: 12 ms

Response

Headers ⁶

Cookies

Results

Docs

```
1  [  
2    {  
3      "id": "6899ed759774a6b9ae8b0226",  
4      "email": "anderson@email.com",  
5      "nome": "Anderson",  
6      "idade": "51"  
7    },  
8    {  
9      "id": "6899f68295b2592bf9afe0ae",  
10     "email": "maria@email.com",  
11     "nome": "Maria",  
12     "idade": "35"  
13   }  
14  ]
```

NA PRÓXIMA AULA VAMOS CRIAR AS ROTAS
PARA ATUALIZAR E PARA REMOVER UM
USUÁRIO.