

SENAC
Campus Santo Amaro
TADS - Análise Desenvolvimento de Sistemas



PW - Programação Web
ADO #04
Beatriz Gomes Feliciano
Carlos Veríssimo

São Paulo, 2022

- `/prisma/schema.prisma`

```
// This is your Prisma schema file,  
// learn more about it in the docs: https://pris.ly/d/prisma-schema  
  
generator client {  
  provider = "prisma-client-js"  
}  
  
datasource db {  
  provider = "mysql"}
```

```

url      = env("DATABASE_URL")
}

model User {
  id String @id @default(uuid())
  nomeUser String
  senha String
  perfilUser String
  criacao DateTime? @default(now())
  alteracao DateTime? @default(now())
  @@map("users")
}

```

- prisma/migrations/ 20221130115836_tabela/ [migration.sql](#)

```

-- CreateTable
CREATE TABLE `users` (
  `id` VARCHAR(191) NOT NULL,
  `nomeUser` VARCHAR(191) NOT NULL,
  `senha` VARCHAR(191) NOT NULL,
  `perfilUser` VARCHAR(191) NOT NULL,
  `criacao` DATETIME(3) NULL DEFAULT CURRENT_TIMESTAMP(3),
  `alteracao` DATETIME(3) NULL DEFAULT CURRENT_TIMESTAMP(3),

  PRIMARY KEY (`id`)
) DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

```

```

src > prisma > TS index.ts > ...
1   import { PrismaClient } from "@prisma/client";
2   //
3   const prismaClient = new PrismaClient();
4   //
5   export default prismaClient;
6

```

Codigo html

```
index.html > html > body > script > getUsers
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10 <script>
11   const url = "http://localhost:3333/tadosUsers";
12
13   async function getUsers(){
14     await fetch(url)
15       .then(async response => {
16         await response.json().then(async data => {
17           await data.forEach(element => {
18             let h2 = document.getElementById('users');
19             let firstp = document.createElement("p");
20             let secondp = document.createElement("p");
21             firstp.textContent = "Nome: " + element.nomeUser;
22             h2.appendChild(firstp);
23             secondp.textContent = "Perfil: " + element.perfilUser;
24             h2.appendChild(secondp);
25           });
26         });
27       });
28   }
29
30   getUsers();
31 </script>
32
33 <h1 id="users">Lista com os usuarios</h1>
34 </body>
35 </html>
```

Env

```
1 # See the documentation for all the connection string options: https://pris.ly/d/connection-strings
2 # Environment variables declared in this file are automatically made available to Prisma.
3 # See the documentation for more detail: https://pris.ly/d/prisma-schema#accessing-environment-variables-from-the-schema
4 # Prisma supports the native connection string format for PostgreSQL, MySQL, SQLite, SQL Server, MongoDB and CockroachDB.
5 # See the documentation for all the connection string options: https://pris.ly/d/connection-strings
6
7 DATABASE_URL="mysql://root:P@$$w0rd@localhost:4306/nomedobanco?schema=public"
```

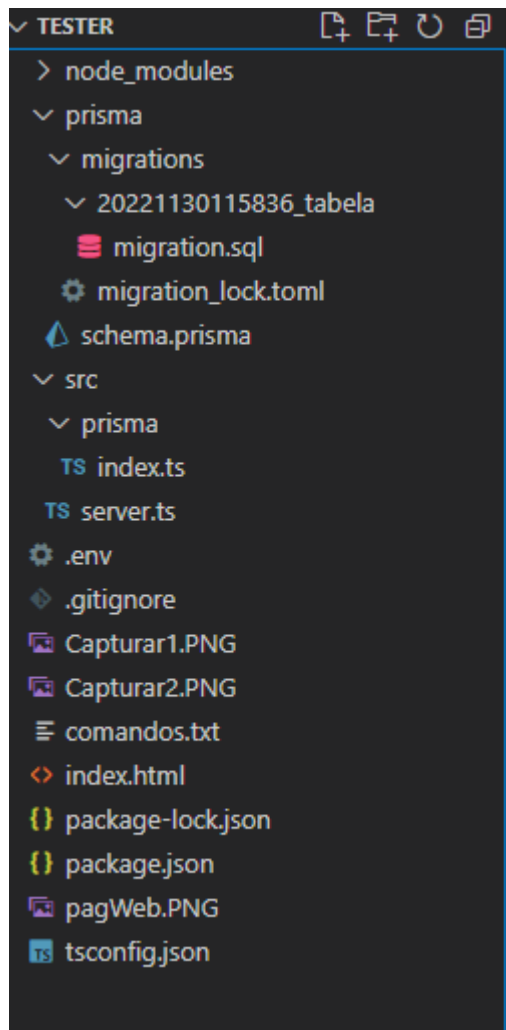
Server

```

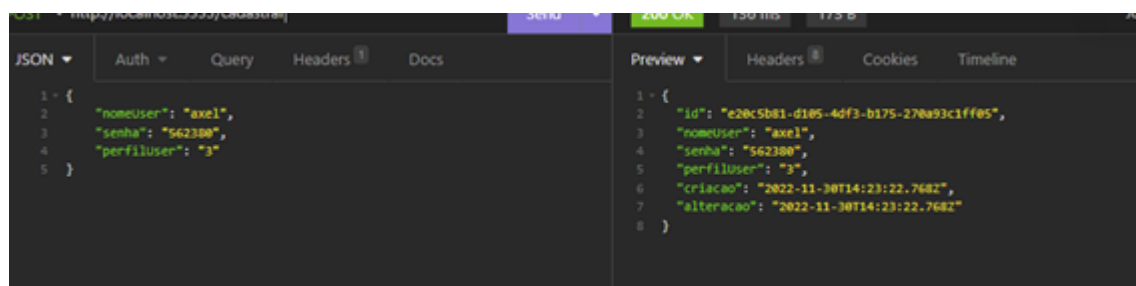
1  import express, { Request, Response } from 'express';
2  import cors from 'cors';
3  import prismaClient from './prisma';
4
5  const app = express();
6  app.use(express.json());
7
8  app.use(cors());
9
10 app.post('/cadastrar', async (req: Request, res: Response) => {
11
12     const { nomeUser, senha, perfilUser } = req.body;
13
14     execute({nomeUser, senha, perfilUser});
15
16     async function execute({nomeUser, senha, perfilUser}: userRequest) {
17
18         const user = await prismaClient.user.create({
19             data: {
20                 nomeUser: nomeUser,
21                 senha: senha,
22                 perfilUser: perfilUser,
23             }
24         });
25
26         return res.status(200).json(user);
27     }
28 })
29
30 app.get('/todosUsers', async (req: Request, res: Response) => {
31
32     const users = await prismaClient.user.findMany();
33
34     return res.status(200).json(users);
35 })
36

```

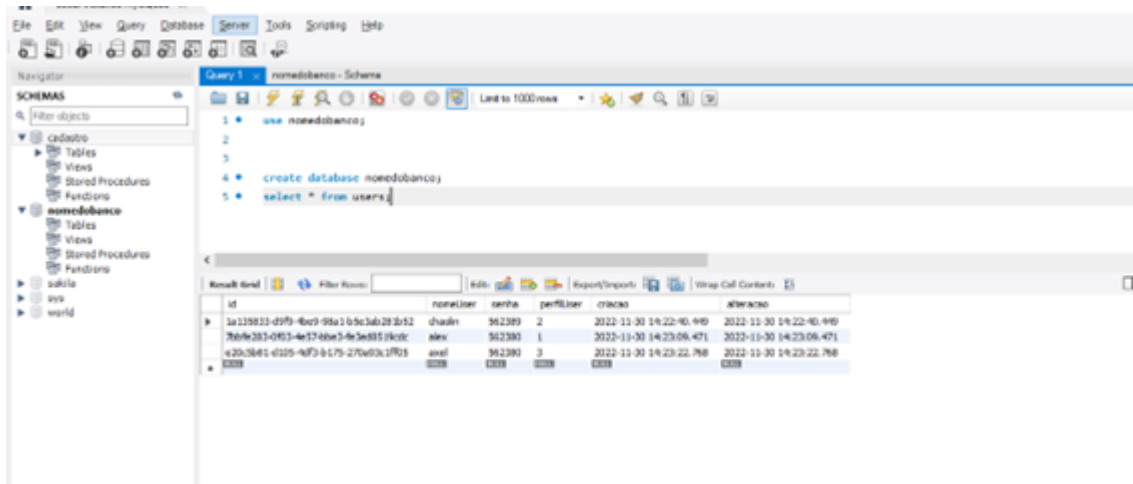
Print arquitetura projeto



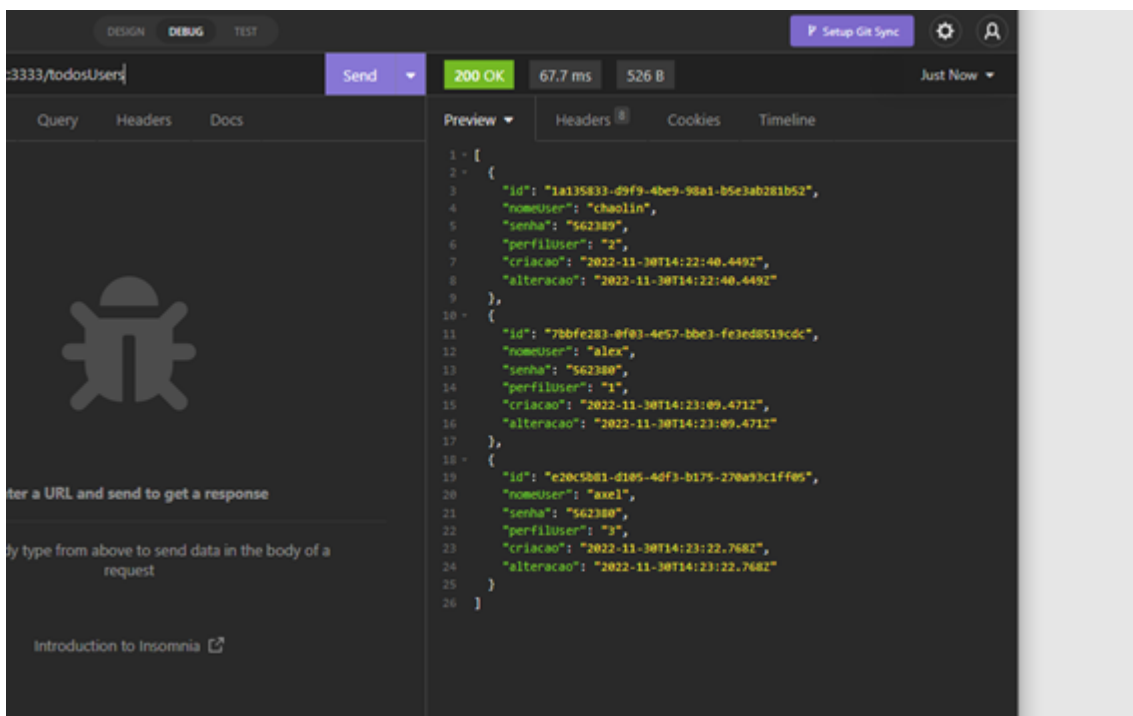
Print cadastro de um usuario



Print workbench Banco



Print API todos os usuarios



Print Pagina HTML

Lista com os usuários

Nome: chaolin

Perfil: 2

Nome: alex

Perfil: 1

Nome: axel

Perfil: 3