Criando minha primeira API nodejs

Pela ótica de um Dev Java

Hello, Devs
sou o Giovanni

Sou desenvolvedor e tenho +4 anos de experiência em engenharia de software, integração de sistemas, transformação digital e modernização de plataformas legadas para cloud AWS.

Minha Timeline

Educacional

SENAI

FAM

Profissional

Itaú

Pessoal

Agenda de hoje

Diferenças entre Java x JS

Diferenças de ambientes de projetos

Diferenças de config de ambiente

Diferenças de criação de projetos

Diferenças de estrutura de projetos

Análise de código

Antes de começar!

Java

1. Java foi criada pela Sun Microsystems em 1995 e atualmente é mantida pela Oracle.

2. Definida como uma linguagem de programação orientada a objetos.

3. A principal função do Java é construir aplicações distribuídos em múltiplas plataformas, como mac, windows e linux.

4. Java Virtual Machine (JVM) é quem interpreta o bytecode para executar como um programa Java,

JavaScript

1. Jogada de marketing feita para chamar atenção para o JavaScript através da fama que o Java já tinha conquistado naquele momento.

2. A principal finalidade do JavaScript é adicionar interação às páginas, tornando possível a interação entre usuários e as aplicações Web e evitando que uma página seja meramente estática ao mostrar informações.

3. A principal vantagem do JavaScript é a possibilidade de uso em todas essas camadas da aplicação.

4. O JS, segundo a Pesquisa de Devs do Stack Overflow de 2022, é a mais popular no mundo.

Diferenças entre Java x JS

Linguagem interpretada ou compilada

- JavaScript é uma linguagem de programação interpretada, ou seja, pode ser lida e traduzida ao mesmo tempo que o programa está sendo executado. Enquanto isso, o Java é uma linguagem compilada: passa por uma JVM (Java Virtual Machine) de forma que ela é interpretada e traduzida.

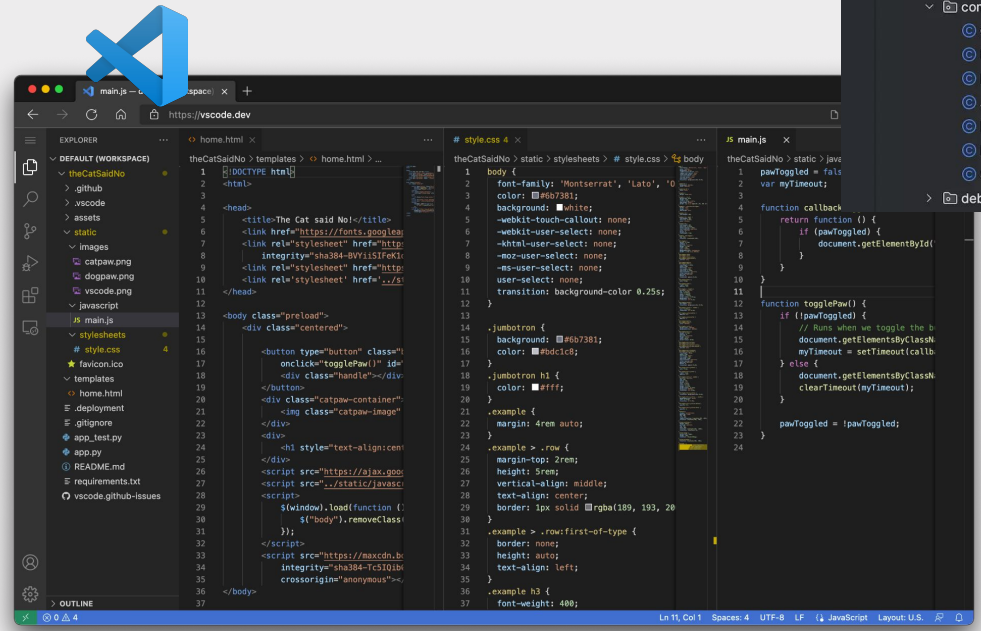
Tipagem

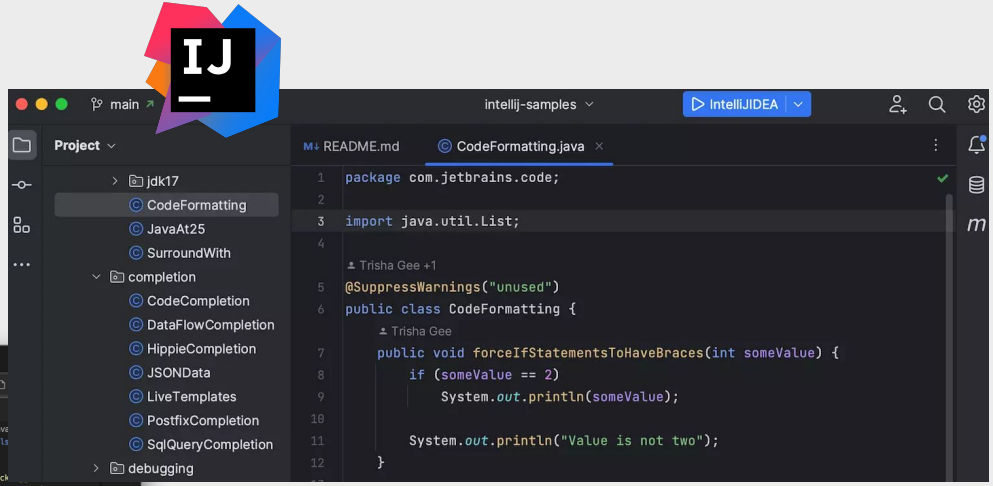
- Outra diferença é que JavaScript é uma linguagem weakly-typed, enquanto Java é uma linguagem strongly-typed. Estas definições estão relacionadas com quão estreitas são as regras de escrita de ambas.

Backend ou frontend

- Outro ponto que as diferencia é o fato delas não serem utilizadas com o mesmo propósito. Enquanto Javascript é uma linguagem recomendada para apps, websites e front-end, Java é uma linguagem mais voltada para back-end e aplicações independentes de outro software e hardware.

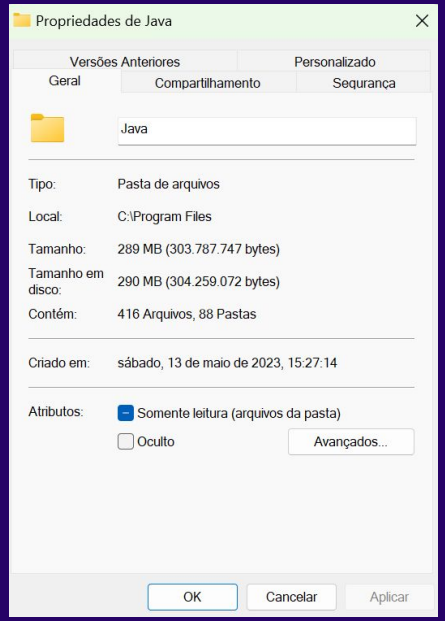
Diferenças de ambientes de projetos

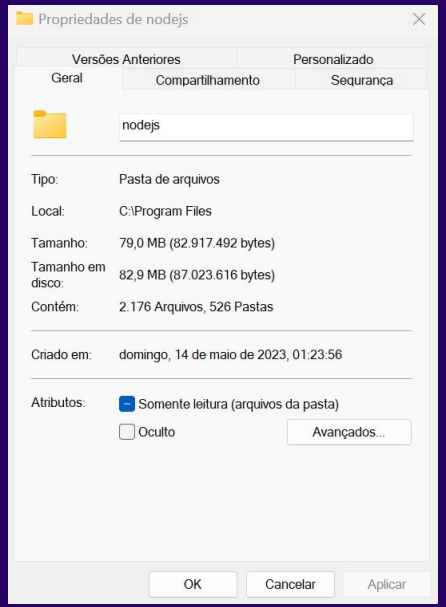




- O Vs Code é um editor de código
- O IntelliJ é uma IDE

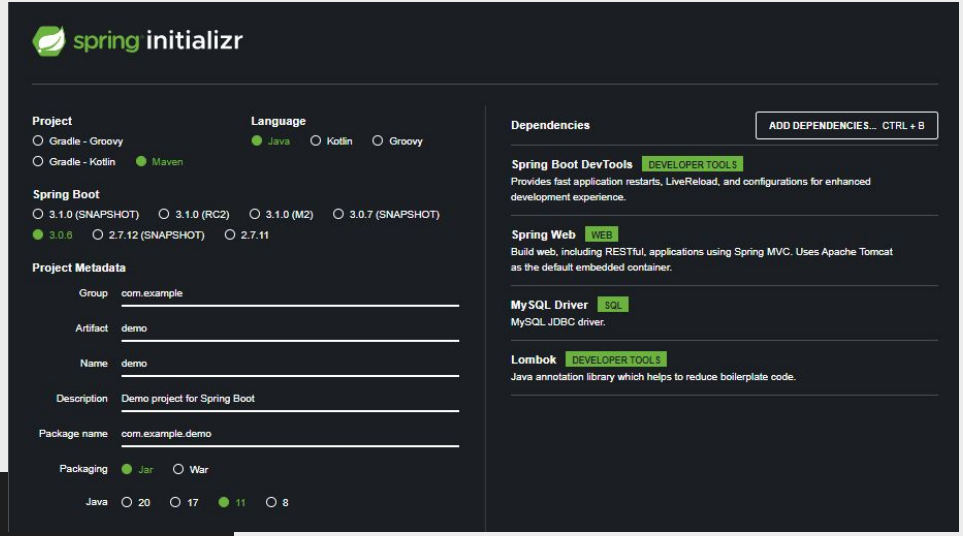
Diferenças de config de ambiente

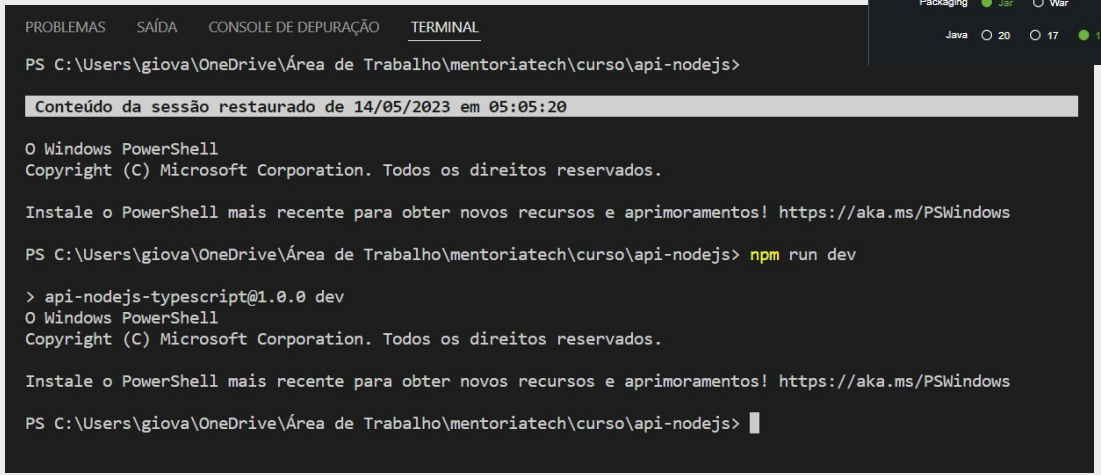




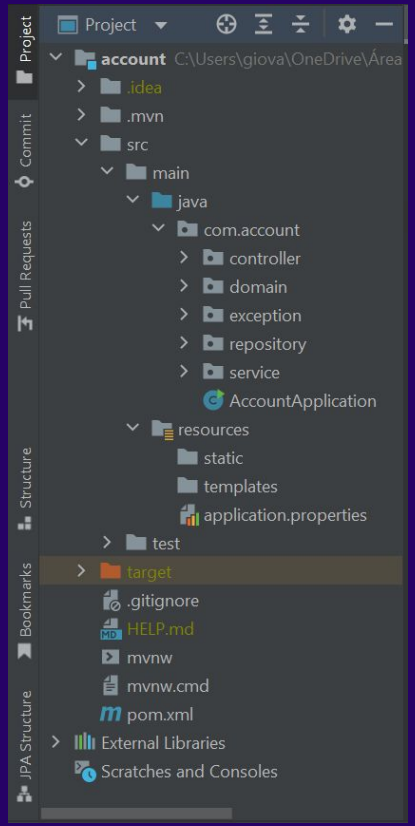
- O Java tem 289 MB enquanto o JS tem 79 MB.
- O tamanho do disco do Java é de 290 MB já o do JS é de 83 MB.
- Java dividido em +400 arquivos e o JS em +2100 arquivos (node modules)

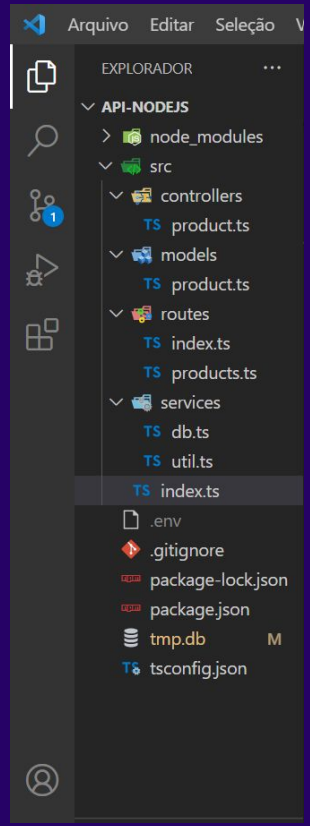
Diferenças de criação de projetos





Diferenças de estrutura de projeto

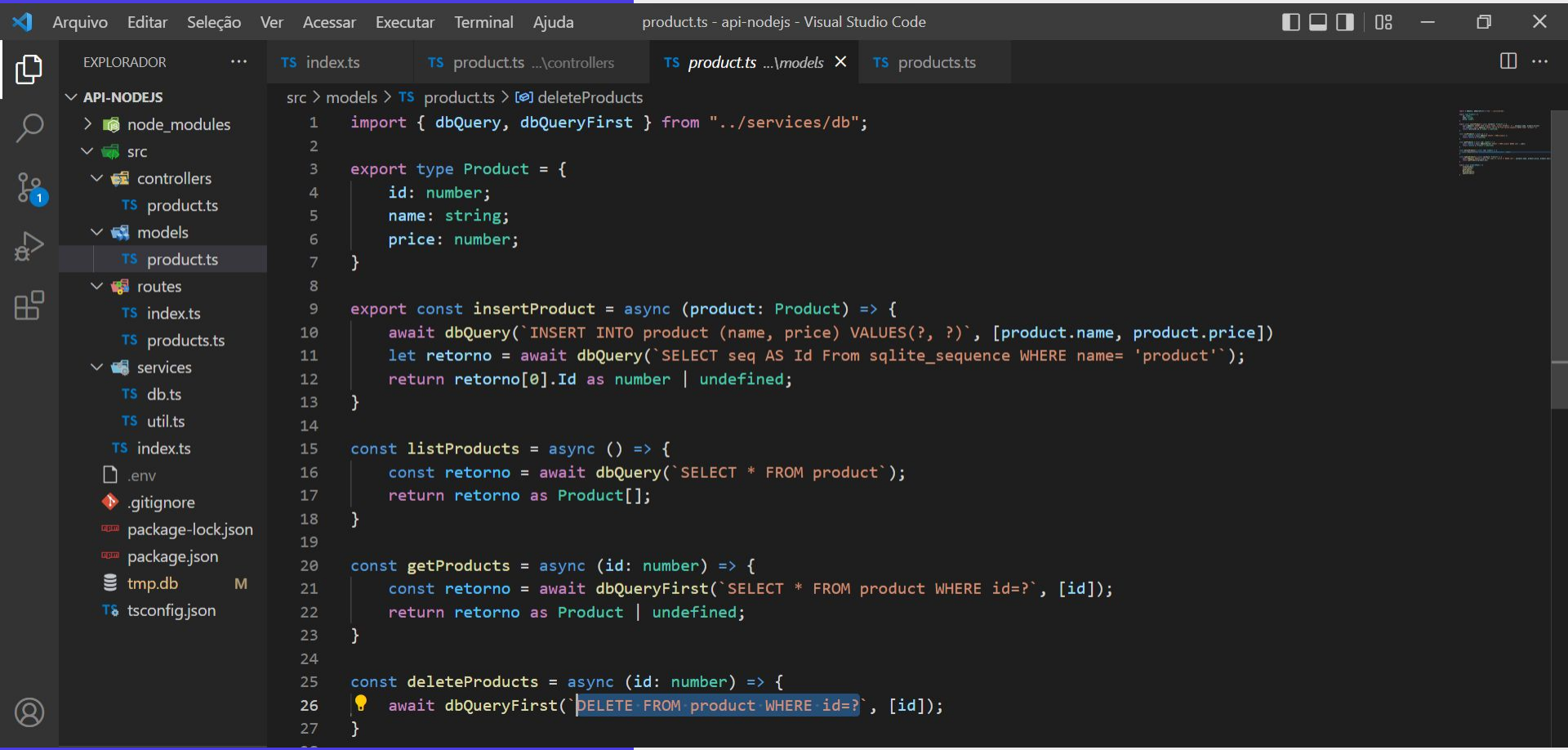


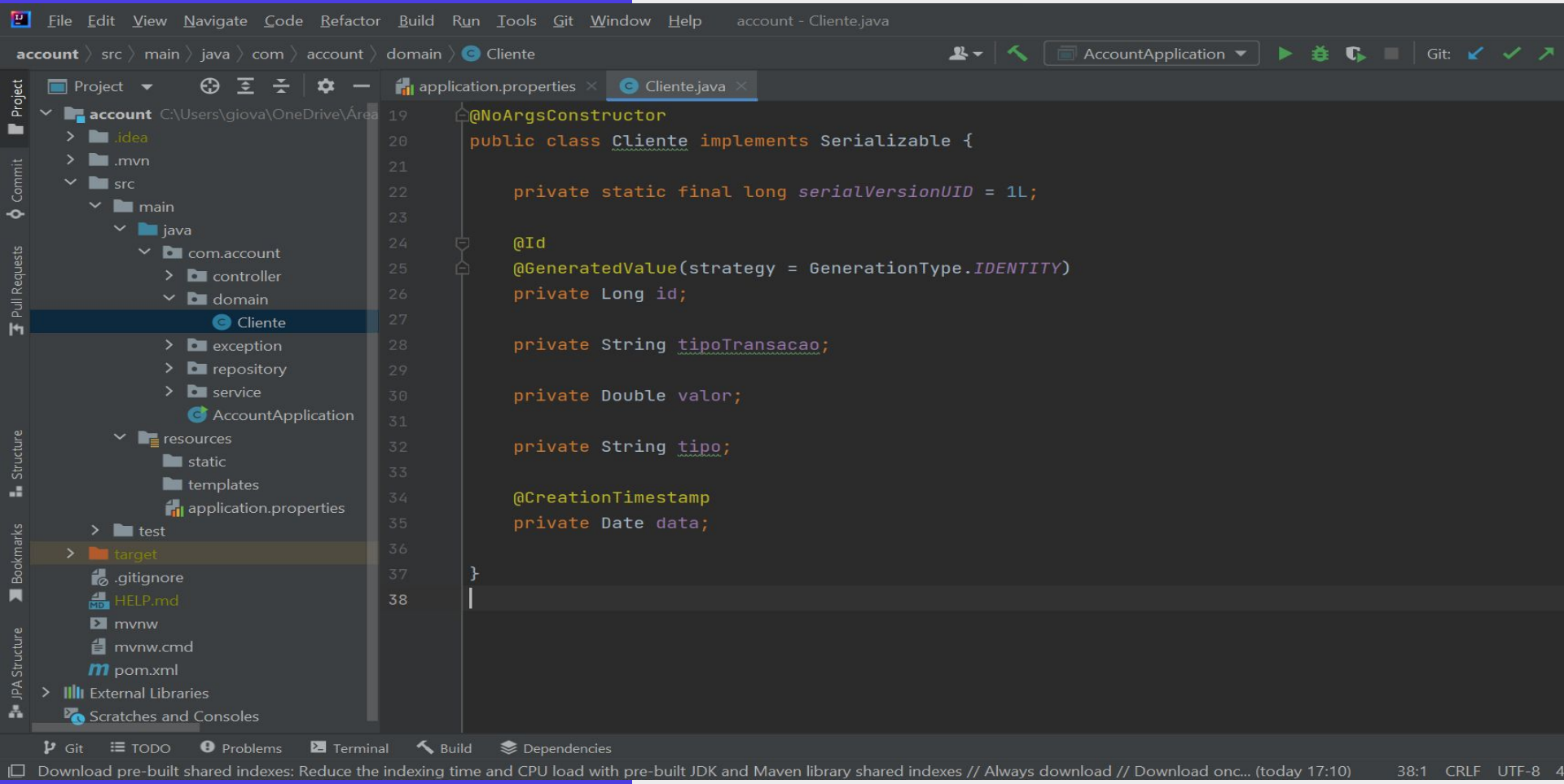


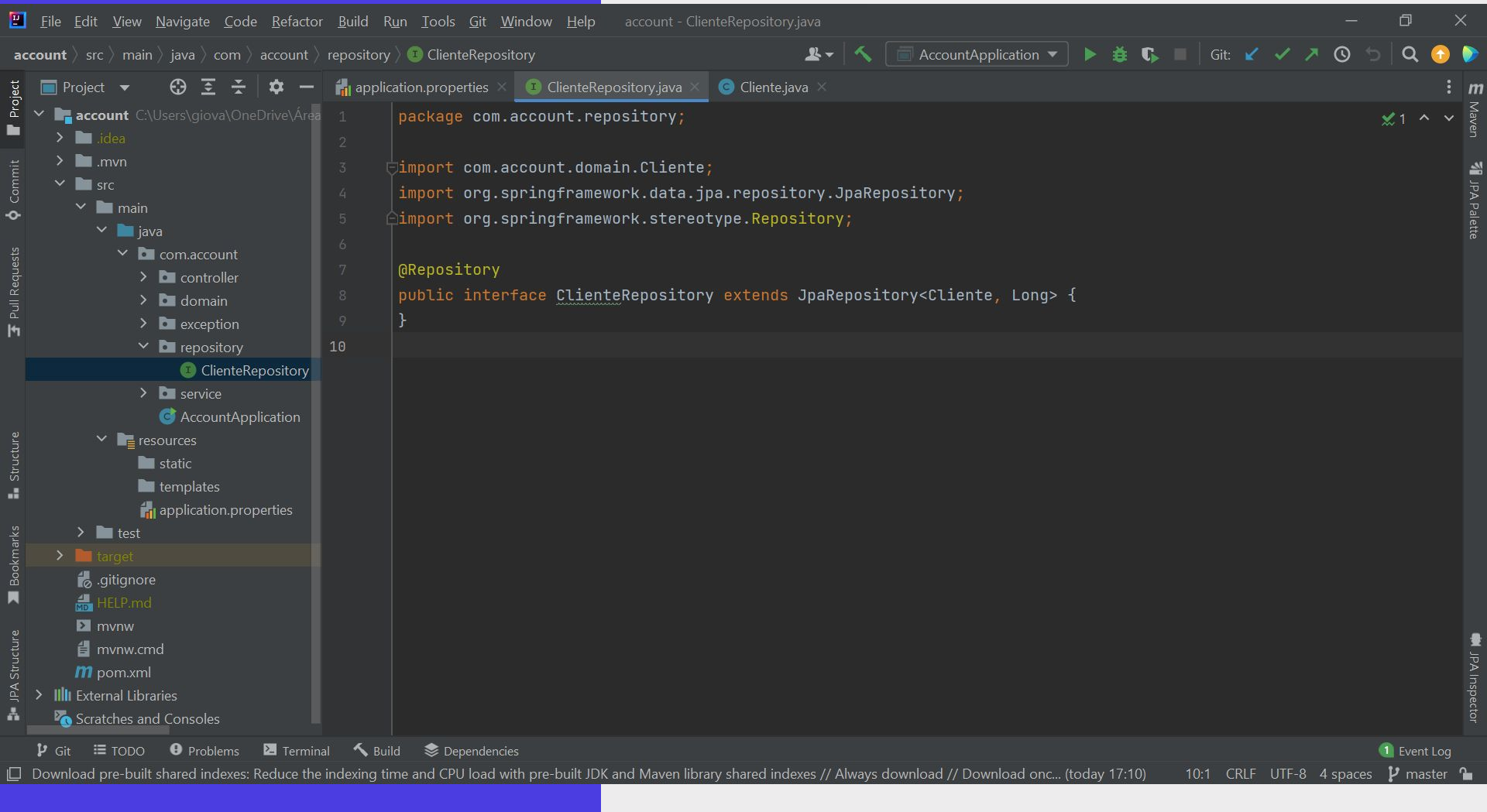
- No java realizamos a estrutura de projetos com a extensão .class onde utilizamos a OOP tradicional.

- No Js realizamos a criação da estrutura com a extensão .TS utilizando a OOP mais flexível

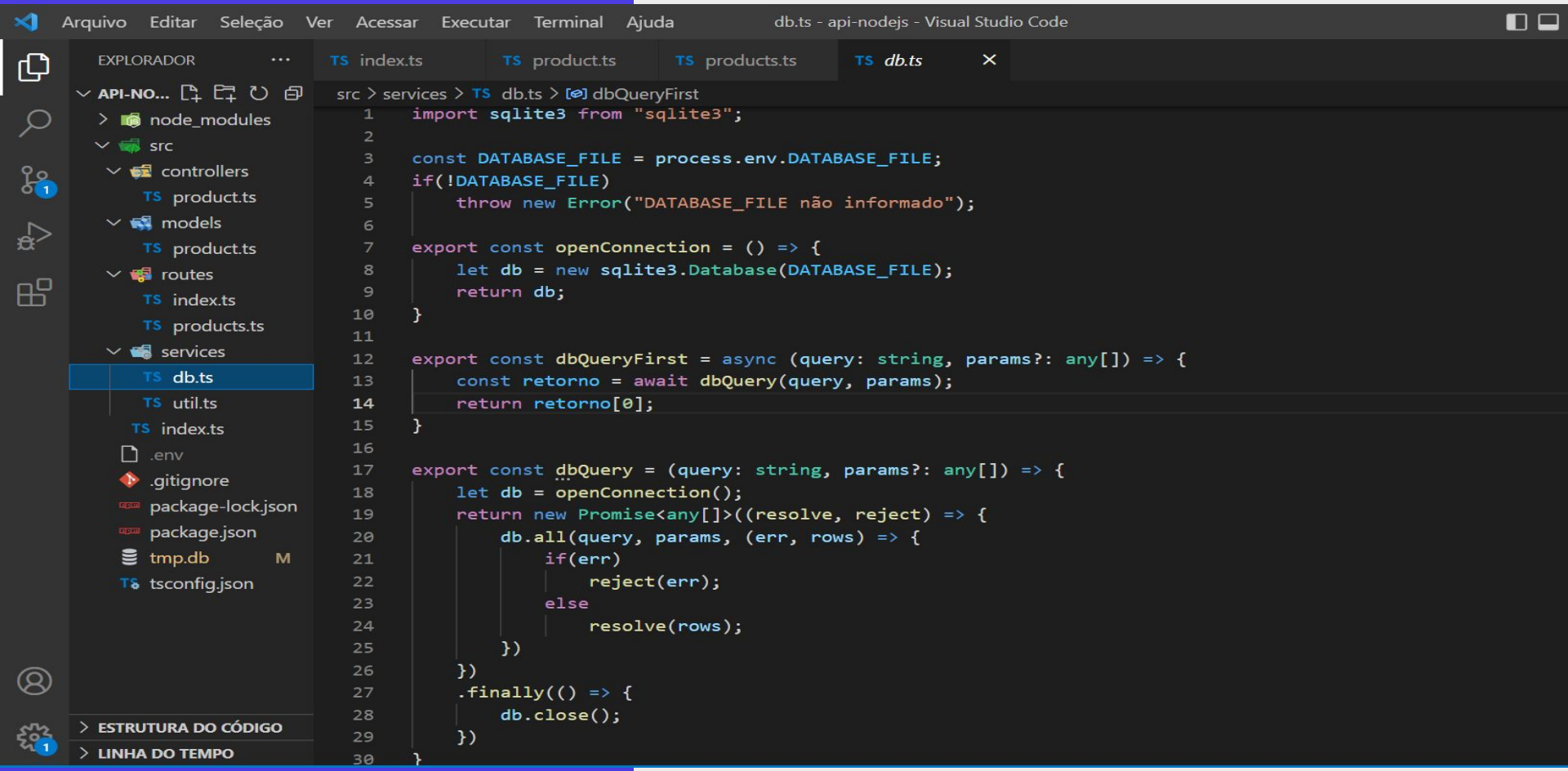
Análise de código - Model

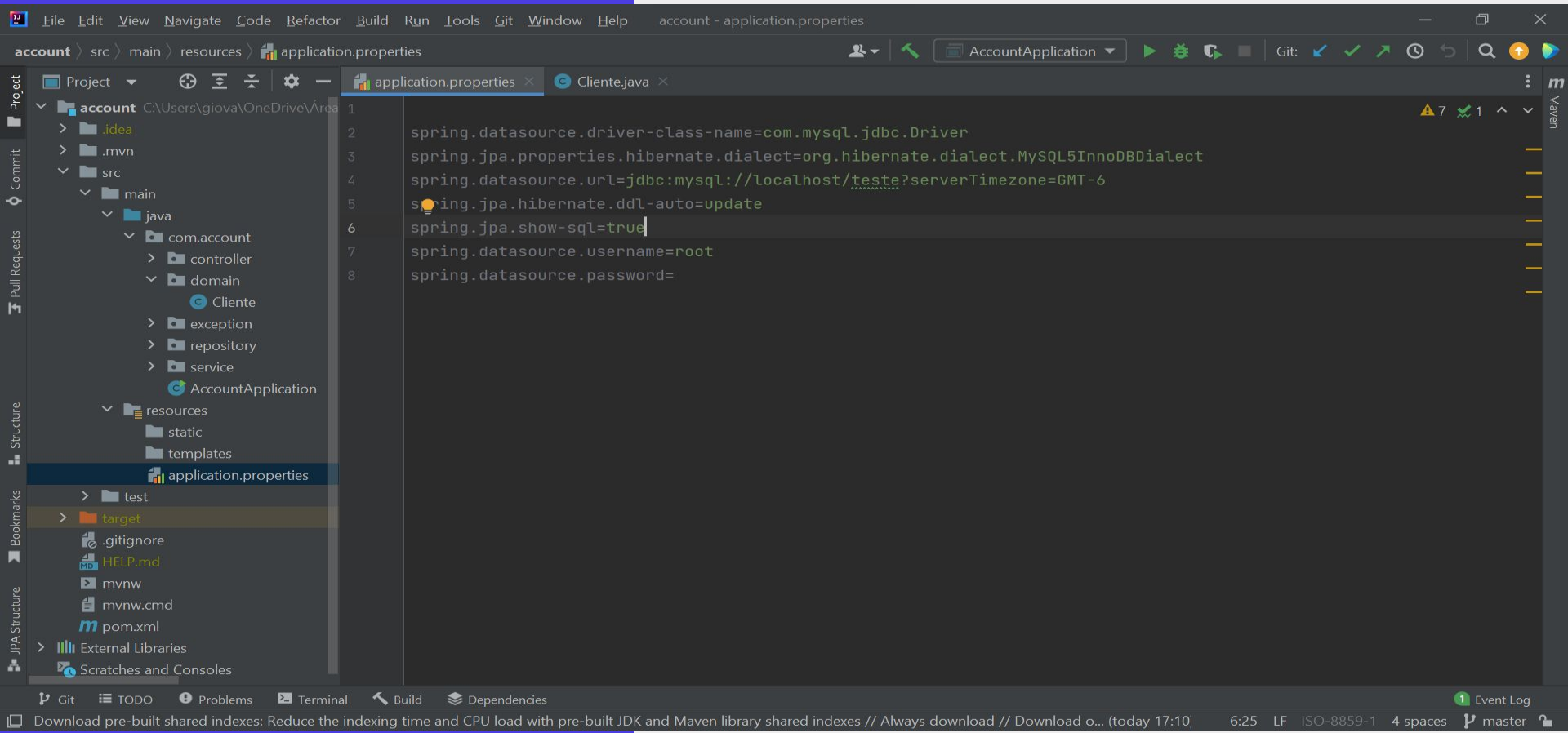

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
product.ts - api-nodejs - Visual Studio Code
EXPLORADOR
API-NODEJS
node_modules
src
controllers
product.ts
models
product.ts
routes
index.ts
products.ts
services
db.ts
util.ts
index.ts
.env
.gitignore
package-lock.json
package.json
tmp.db M
tsconfig.json
index.ts
product.ts ...\controllers
product.ts ...\models
products.ts
src > models > product.ts > deleteProducts
import { dbQuery, dbQueryFirst } from "../services/db";
export type Product = {
id: number;
name: string;
price: number;
}
export const insertProduct = async (product: Product) => {
await dbQuery(`INSERT INTO product (name, price) VALUES(?, ?)`, [product.name, product.price])
let retorno = await dbQuery(`SELECT seq AS Id From sqlite_sequence WHERE name= 'product'`);
return retorno[0].Id as number | undefined;
}
const listProducts = async () => {
const retorno = await dbQuery(`SELECT * FROM product`);
return retorno as Product[];
}
const getProducts = async (id: number) => {
const retorno = await dbQueryFirst(`SELECT * FROM product WHERE id=?`, [id]);
return retorno as Product | undefined;
}
const deleteProducts = async (id: number) => {
await dbQueryFirst(`DELETE FROM product WHERE id=?`, [id]);
}


File Edit View Navigate Code Refactor Build Run Tools Git Window Help account - Cliente.java
account src main java com account domain Cliente
AccountApplication
Git:
Project
account C:\Users\giova\OneDrive\Área
.idea
.mvn
src
main
java
com.account
controller
domain
Cliente
exception
repository
service
AccountApplication
resources
static
templates
application.properties
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
application.properties
Cliente.java
@NoArgsConstructor
public class Cliente implements Serializable {
private static final long serialVersionUID = 1L;
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;
private String tipoTransacao;
private Double valor;
private String tipo;
@CreationTimestamp
private Date data;
}
Git TODO Problems Terminal Build Dependencies
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // Always download // Download onc... (today 17:10)
38:1 CRLF UTF-8

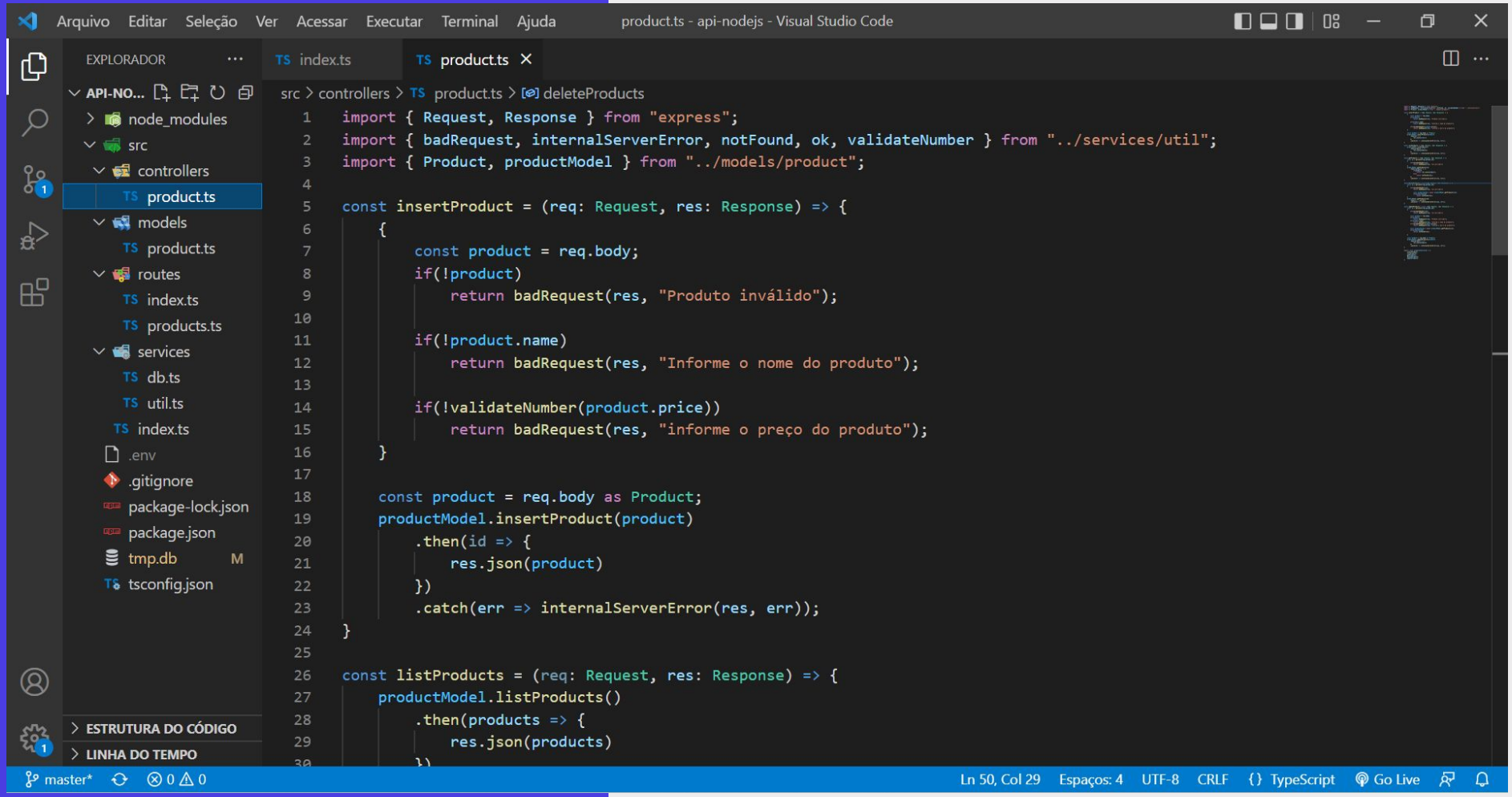

File Edit View Navigate Code Refactor Build Run Tools Git Window Help
account - ClienteRepository.java
account src main java com account repository ClienteRepository
AccountApplication
Git:
Project
account C:\Users\giova\OneDrive\Área
.idea
.mvn
src
main
java
com.account
controller
domain
exception
repository
ClienteRepository
service
AccountApplication
resources
static
templates
application.properties
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
application.properties
ClienteRepository.java
Cliente.java
1 package com.account.repository;
2
3 import com.account.domain.Cliente;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository
8 public interface ClienteRepository extends JpaRepository<Cliente, Long> {
9 }
10
Maven
JPA Palette
Project
Commit
Pull Requests
Structure
Bookmarks
JPA Structure
JPA Inspector
Git
TODO
Problems
Terminal
Build
Dependencies
Event Log
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // Always download // Download onc... (today 17:10)
10:1
CRLF
UTF-8
4 spaces
master

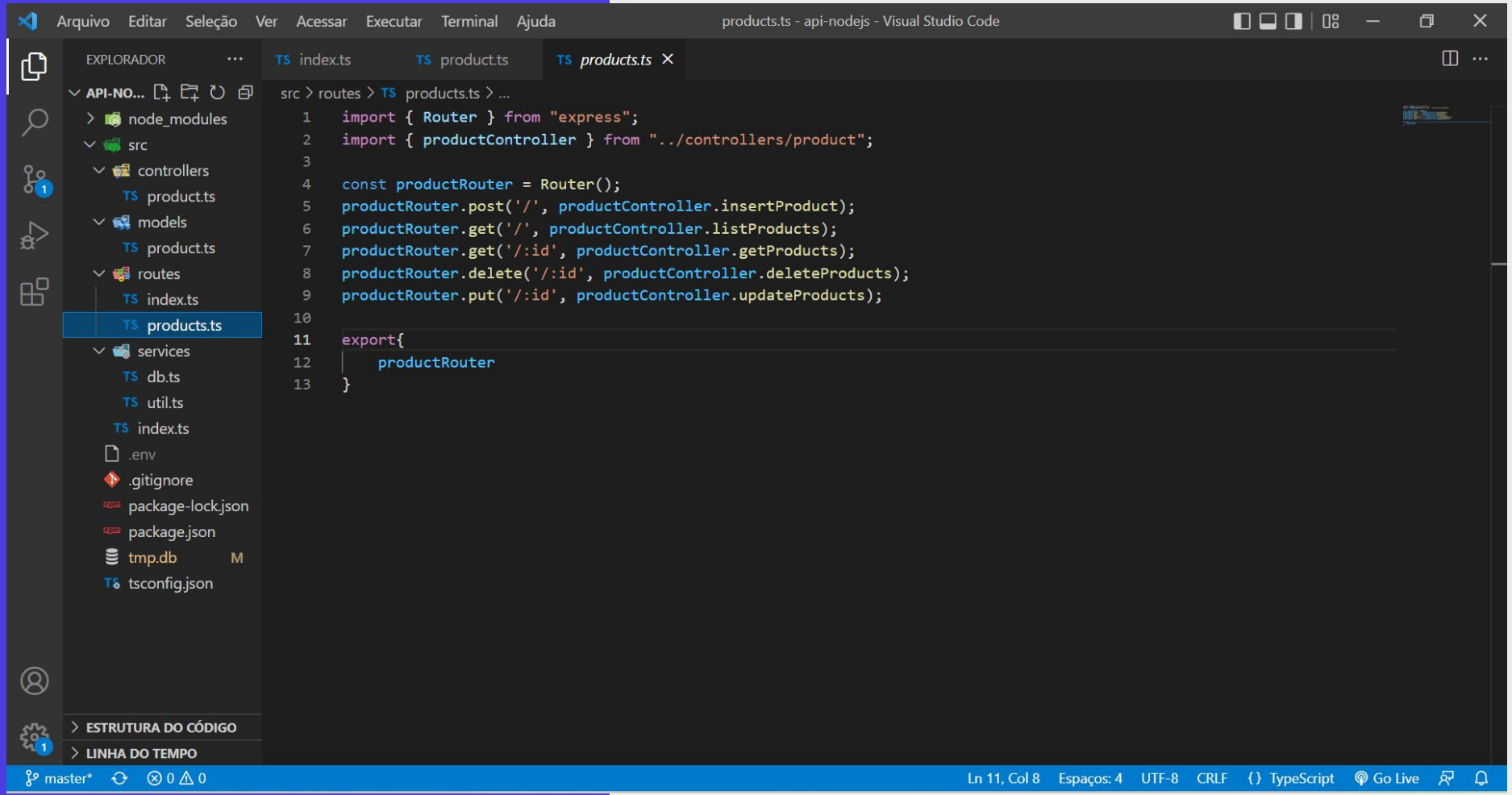
Análise de código - Conexão DB

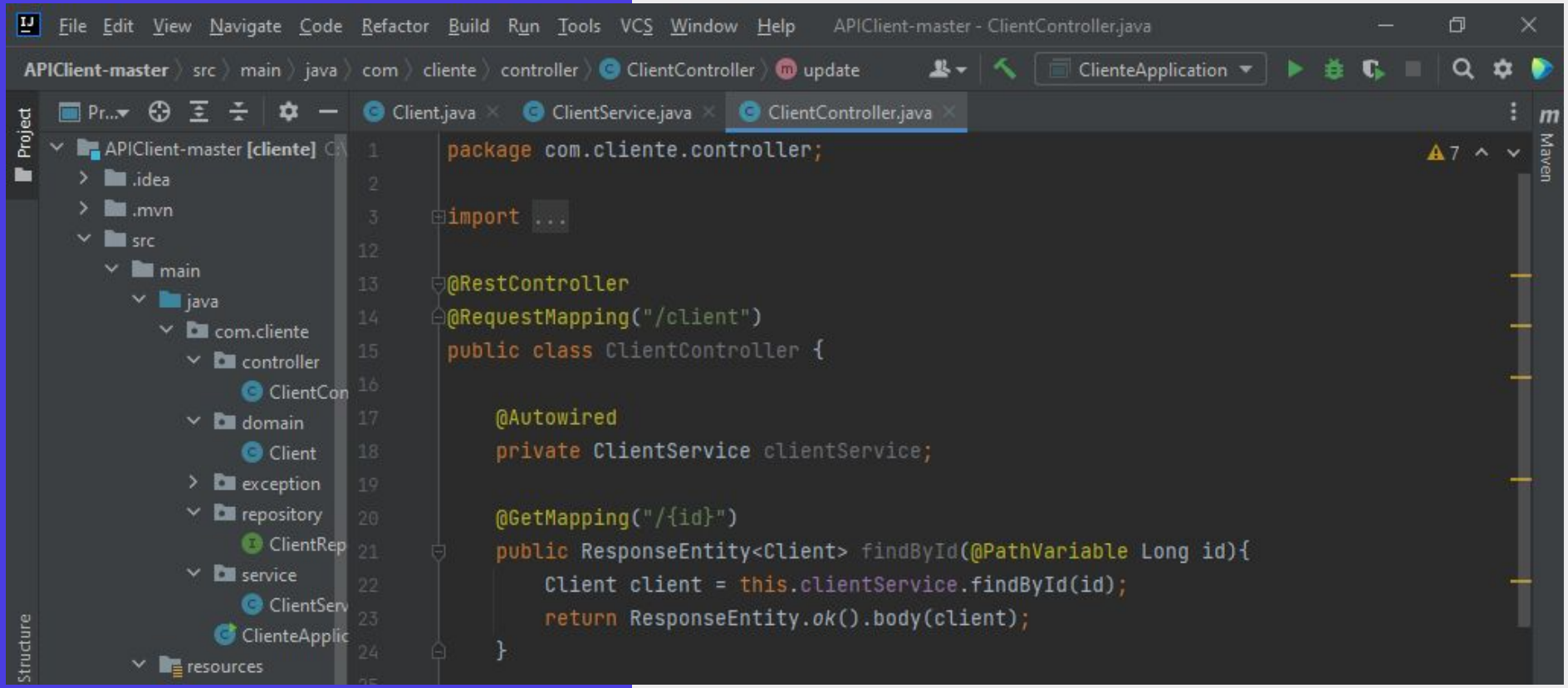

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
db.ts - api-nodejs - Visual Studio Code
EXPLORADOR
API-NO...
node_modules
src
controllers
product.ts
models
product.ts
routes
index.ts
products.ts
services
db.ts
util.ts
index.ts
.env
.gitignore
package-lock.json
package.json
tmp.db M
tsconfig.json
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
index.ts
product.ts
products.ts
db.ts
src > services > db.ts > dbQueryFirst
1 import sqlite3 from "sqlite3";
2
3 const DATABASE_FILE = process.env.DATABASE_FILE;
4 if(!DATABASE_FILE)
5 throw new Error("DATABASE_FILE não informado");
6
7 export const openConnection = () => {
8 let db = new sqlite3.Database(DATABASE_FILE);
9 return db;
10 }
11
12 export const dbQueryFirst = async (query: string, params?: any[]) => {
13 const retorno = await dbQuery(query, params);
14 return retorno[0];
15 }
16
17 export const dbQuery = (query: string, params?: any[]) => {
18 let db = openConnection();
19 return new Promise<any[]>((resolve, reject) => {
20 db.all(query, params, (err, rows) => {
21 if(err)
22 reject(err);
23 else
24 resolve(rows);
25 })
26 })
27 .finally(() => {
28 db.close();
29 })
30 }


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
account - application.properties
account > src > main > resources > application.properties
AccountApplication
Git:
Project
account C:\Users\giova\OneDrive\Área
.idea
.mvn
src
main
java
com.account
controller
domain
Cliente
exception
repository
service
AccountApplication
resources
static
templates
application.properties
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
application.properties
Cliente.java
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
spring.datasource.url=jdbc:mysql://localhost/teste?serverTimezone=GMT-6
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.datasource.username=root
spring.datasource.password=
Git
TODO
Problems
Terminal
Build
Dependencies
Event Log
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // Always download // Download o... (today 17:10
6:25
LF
ISO-8859-1
4 spaces
master

Análise de código Controller & Rotas

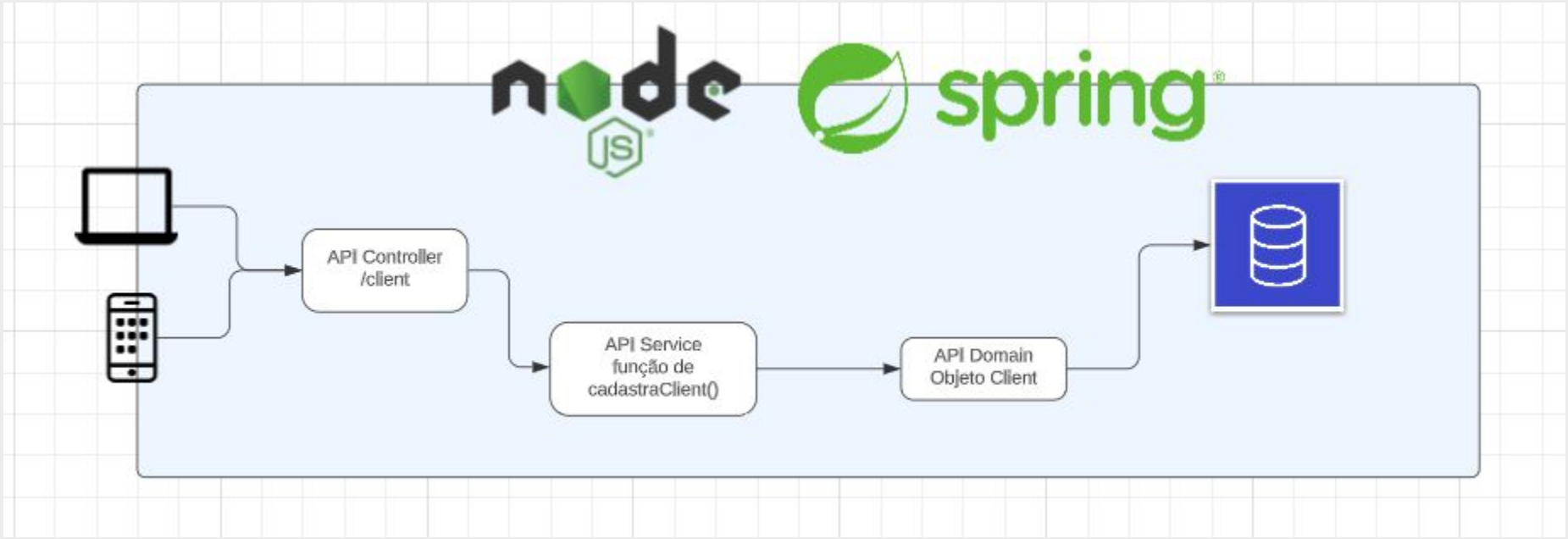

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
product.ts - api-nodejs - Visual Studio Code
EXPLORADOR
API-NO...
node_modules
src
controllers
product.ts
models
product.ts
routes
index.ts
products.ts
services
db.ts
util.ts
index.ts
.env
.gitignore
package-lock.json
package.json
tmp.db M
tsconfig.json
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
index.ts
product.ts
src > controllers > product.ts > deleteProducts
import { Request, Response } from "express";
import { badRequest, internalServerError, notFound, ok, validateNumber } from "../services/util";
import { Product, productModel } from "../models/product";
const insertProduct = (req: Request, res: Response) => {
{
const product = req.body;
if(!product)
return badRequest(res, "Produto inválido");
if(!product.name)
return badRequest(res, "Informe o nome do produto");
if(!validateNumber(product.price))
return badRequest(res, "informe o preço do produto");
}
const product = req.body as Product;
productModel.insertProduct(product)
.then(id => {
res.json(product)
})
.catch(err => internalServerError(res, err));
}
const listProducts = (req: Request, res: Response) => {
productModel.listProducts()
.then(products => {
res.json(products)
master*
0 0
Ln 50, Col 29
Espaços: 4
UTF-8
CRLF
TypeScript
Go Live

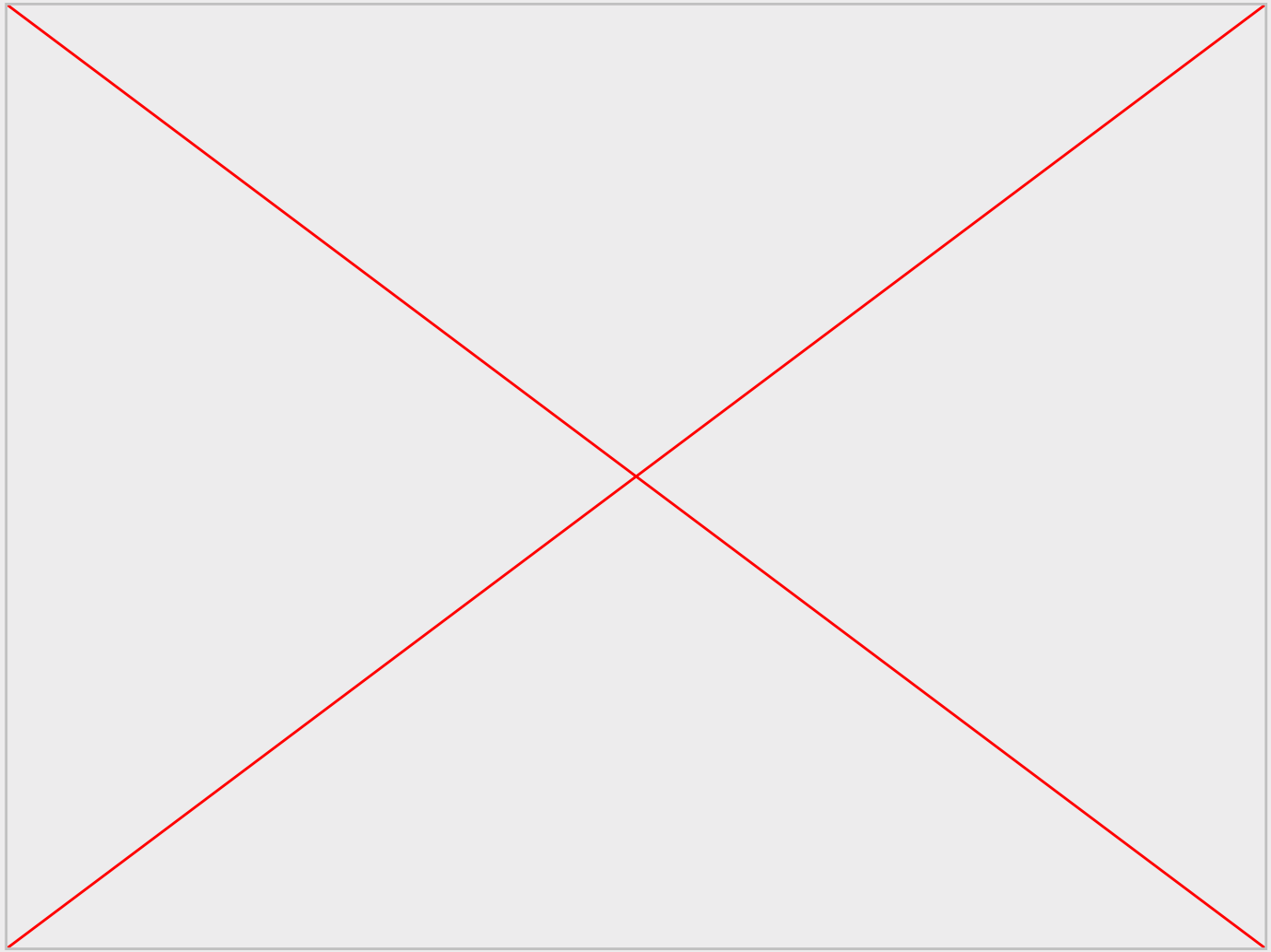

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
products.ts - api-nodejs - Visual Studio Code
EXPLORADOR
index.ts
product.ts
products.ts
src > routes > products.ts > ...
1 import { Router } from "express";
2 import { productController } from "../controllers/product";
3
4 const productRouter = Router();
5 productRouter.post('/', productController.insertProduct);
6 productRouter.get('/', productController.listProducts);
7 productRouter.get('/:id', productController.getProducts);
8 productRouter.delete('/:id', productController.deleteProducts);
9 productRouter.put('/:id', productController.updateProducts);
10
11 export{
12 productRouter
13 }
API-NO...
node_modules
src
controllers
product.ts
models
product.ts
routes
index.ts
products.ts
services
db.ts
util.ts
index.ts
.env
.gitignore
package-lock.json
package.json
tmp.db M
tsconfig.json
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
master*
0 0
Ln 11, Col 8
Espaços: 4
UTF-8
CRLF
TypeScript
Go Live

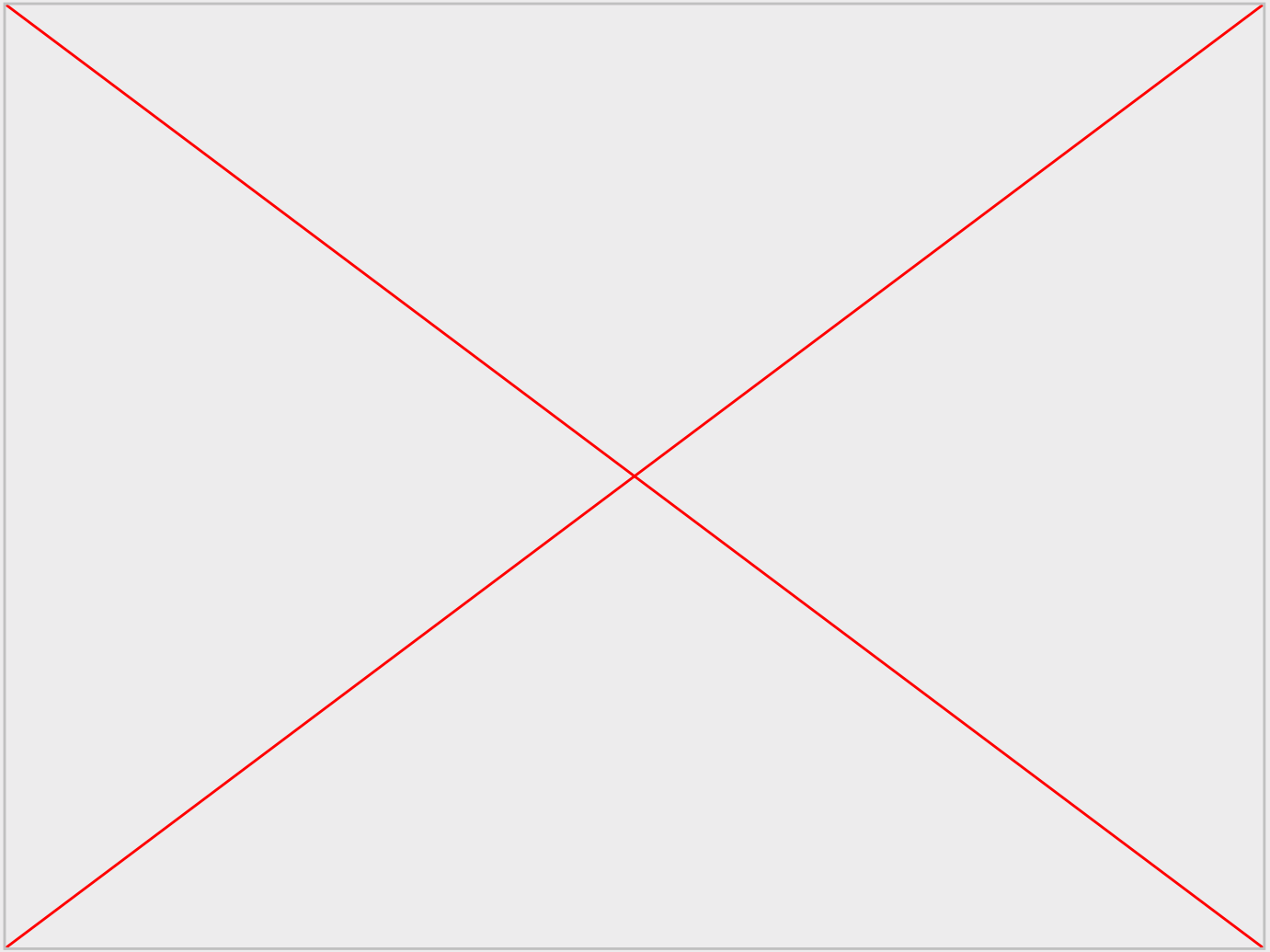

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help APIClient-master - ClientController.java
APIClient-master src main java com cliente controller ClientController update
ClienteApplication
Client.java
ClientService.java
ClientController.java
APIClient-master [cliente]
.idea
.mvn
src
main
java
com.cliente
controller
ClientCon
domain
Client
exception
repository
ClientRep
service
ClientServ
ClienteApplic
resources
Project
Structure
Maven
1 package com.cliente.controller;
3 import ...
13 @RestController
14 @RequestMapping("/client")
15 public class ClientController {
17 @Autowired
18 private ClientService clientService;
20 @GetMapping("/{id}")
21 public ResponseEntity<Client> findById(@PathVariable Long id){
22 Client client = this.clientService.findById(id);
23 return ResponseEntity.ok().body(client);
24 }

Demo

Arquitetura simples da API







Considerações finais!

JAVA

Robusto e
tradicional

JS

Flexível e
moderno

JS

Melhor performance
Curva de aprendizado

JAVA

Mais seguro
Menos personalizável

Obrigadx!

Dúvidas?

Redes Sociais

Instagram

Linkedin