

DESAFIO NEON TECH TALENTS





EQUIPE 1

QUEM SOMOS NÓS?



Adriana Pugarito

Dev. Front-End Junior, seu objetivo é se desenvolver como QA, crescer e trabalhar nessa área.



Amanda Acacio

Dev. Front-End Junior, seu objetivo continuar crescendo na área



Jessica Texeira

Dev. Front-End Junior, seu objetivo é trabalhar e ter sucesso nessa área.



Honer Gelmini

Analista de Governança de TI 1

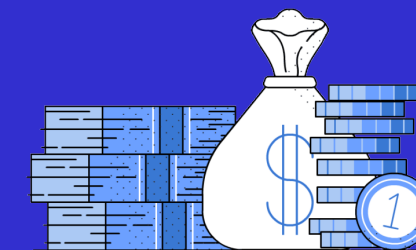


Virginia Zapata

Dev. Back-End Junior, seu objetivo é continuar seus estudos e trabalhar nessa área, implementando o aprendizado.



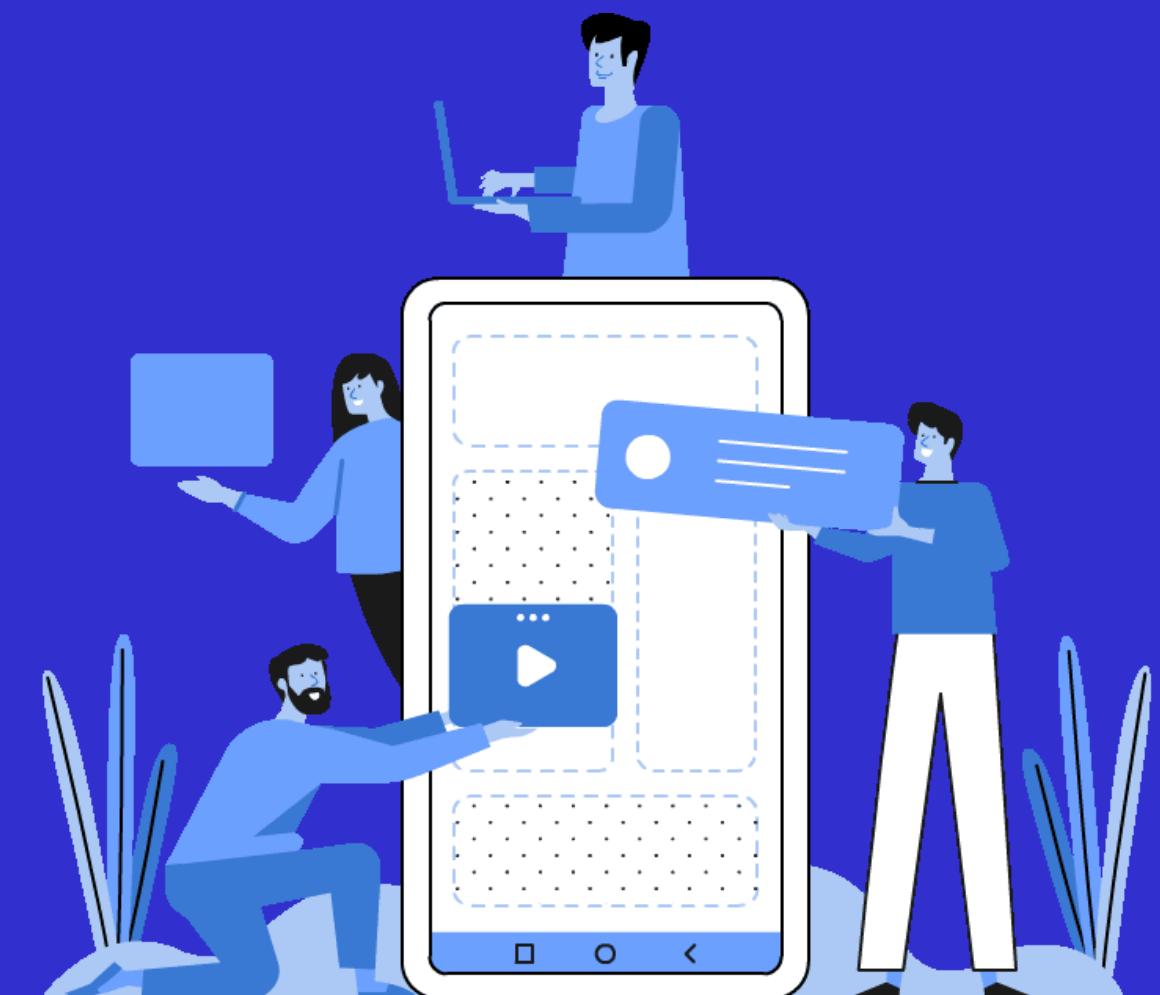
Nosso projeto



Cadê minha grana!

Com a pandemia, é cada vez mais comum as pessoas optarem por não sair de casa para realizar serviços bancários, nossa app é simples e com ela pode ficar de olho em seu dinheiro sem sair de casa.

O que você consegue fazer com Cadê minha grana!



Você consegue fazer:

- Lançamentos de entrada e saída de dinheiro organizando por títulos de contas, despesas e entradas que você possua, administrar seu dinheiro de maneira certa e evitar possíveis gastos desnecessários.
- Você terá acesso a todos os extratos e pagamentos.
- Você não precisa falar com um atendente ou se dirigir a um caixa eletrônico, agora é possível fazer isso de onde você estiver e quando quiser. Basta se cadastrar para criar uma conta e acessar o app, é simple e rápido.



Tecnologia usada no Front-End

Front-End

* Html

* Css

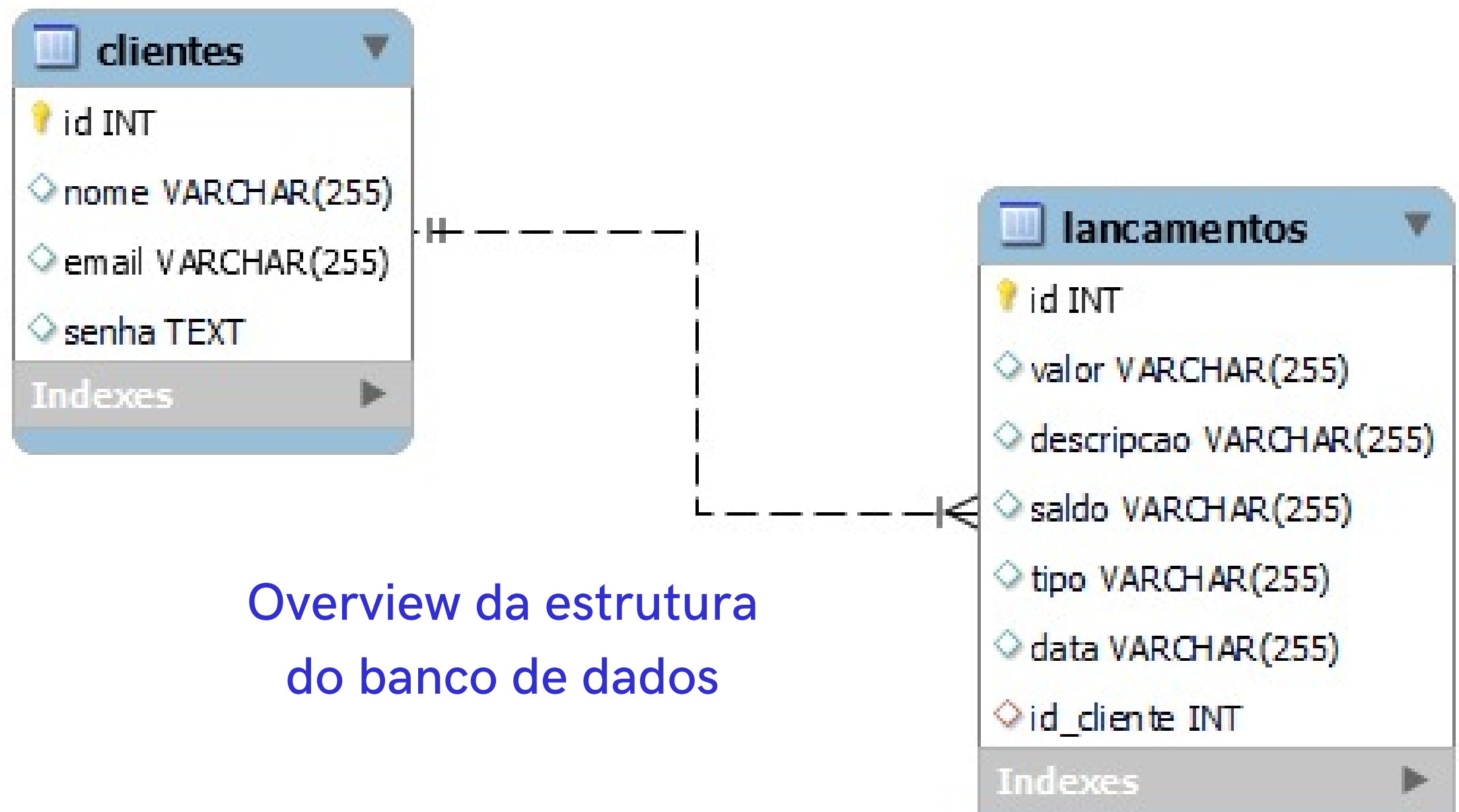
* JavaScript

* Biblioteca Bootstrap

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure for 'GITNEODESAFIO'. The 'Backend' folder is expanded, showing subfolders 'app', 'config', and 'routes', along with files like 'routerCliente.js', 'routerLancam...', 'routesLogin.js', '.env', '.gitignore', '.sequelizerc', 'package-lock.js...', 'package.json', and 'server.js'. The 'server.js' file is selected and its content is displayed in the main editor. The code is a Node.js server using Express.js, with comments in Portuguese. It includes imports for 'express' and 'mongoose', initializes the app, sets up middleware, and defines routes for 'routerCliente', 'routerLancamento', and 'routesLogin'. The server listens on a specified port and logs successful connections.

```
1 const express = require('express'); //importacao do pacote
2 const app = express(); //instanciando express
3 const {connection} = require('./app/database/db');
4 const PORT = process.env.PORT || 5000;
5
6
7 // Middleware
8 // Para poder rellenar el req.body
9 app.use(express.json());
10 app.use(express.urlencoded({ extended: false }));
11
12
13 app.use(require('./routes/routerCliente'))
14 app.use(require('./routes/routerLancamento'))
15 app.use(require('./routes/routesLogin'))
16
17
18
19 app.listen(PORT, function () {
20   console.log(`successfully initialized http://localhost:${PORT}`);
21
22   connection.sync({force: false}).then () => {
23     console.log("Successful Connection");
24   });
25 });
26
```

Tecnologia usada no Back-End



Overview da estrutura
do banco de dados

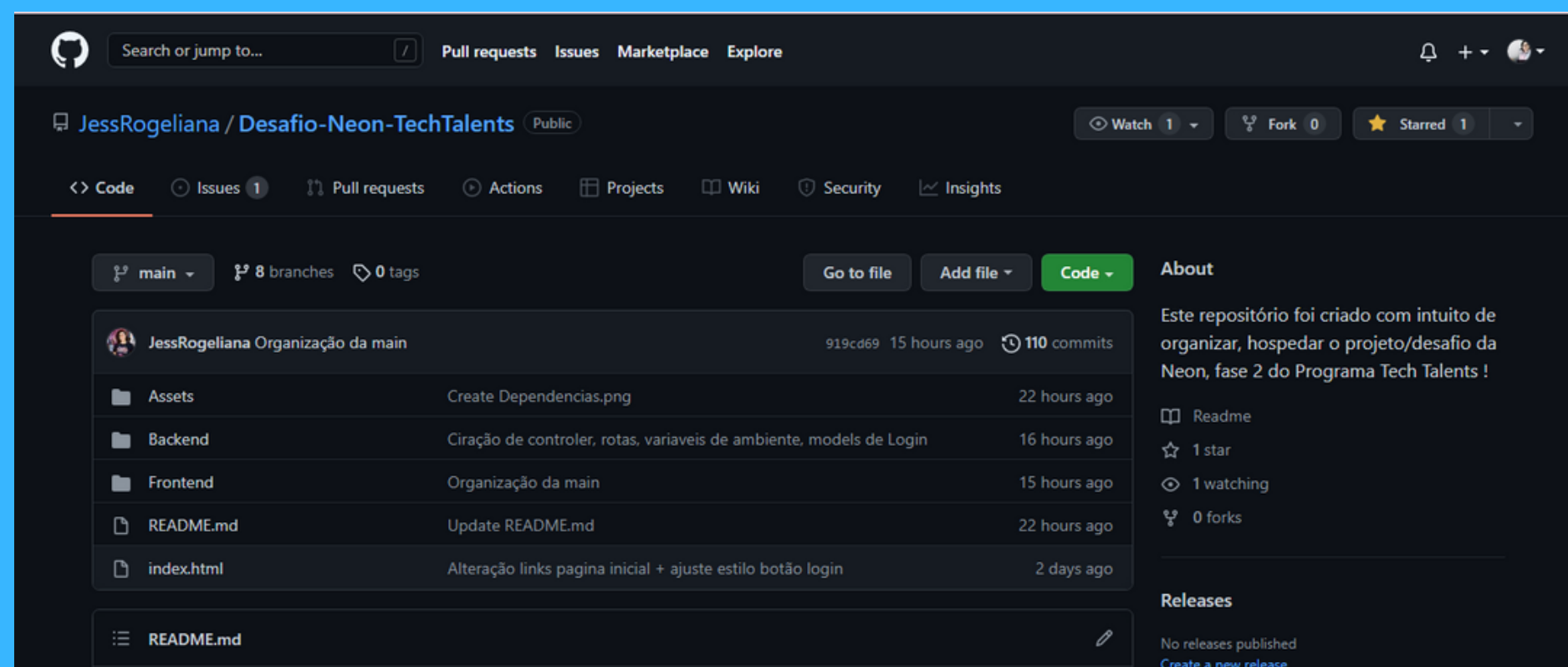
*NodeJs

*MySQL

*Bibliotecas e dependencias:
Cors, Sequelize, Express,
Nodemon, Sequelize-Cli e
Dontenv



Informação de GitHub



Bora testar!!!