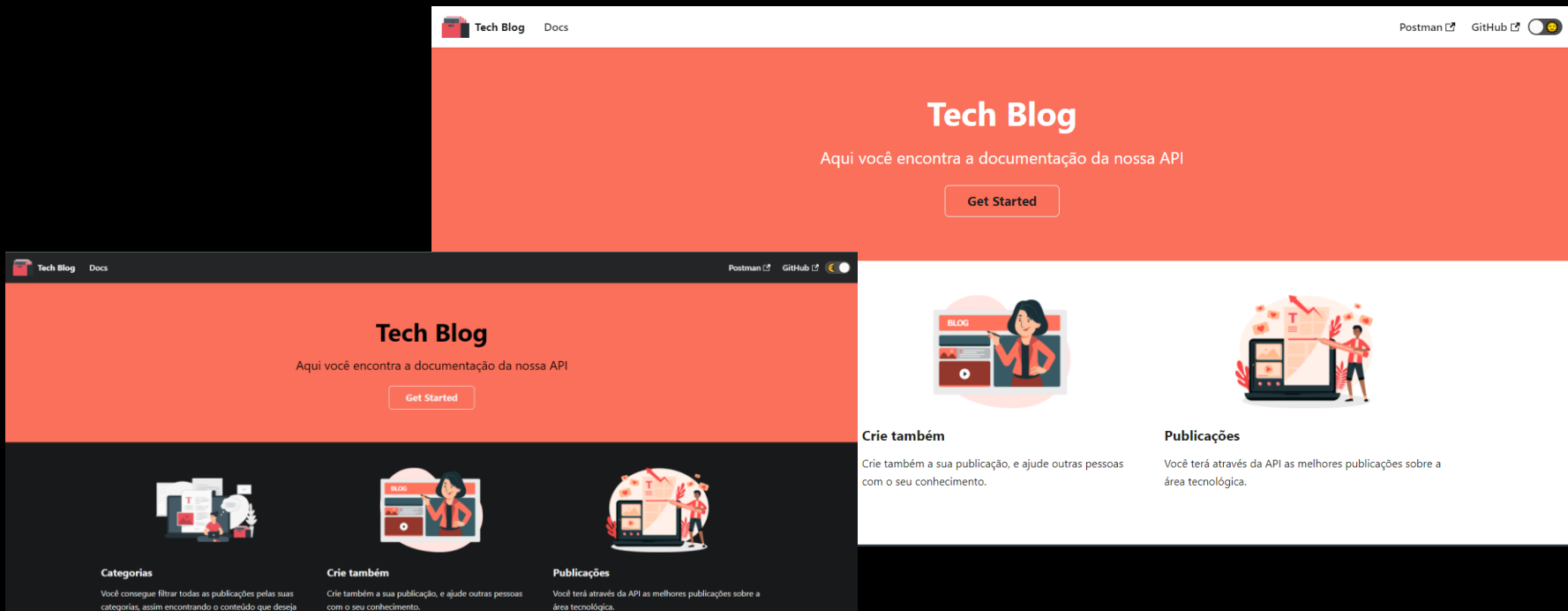


Tech Blog

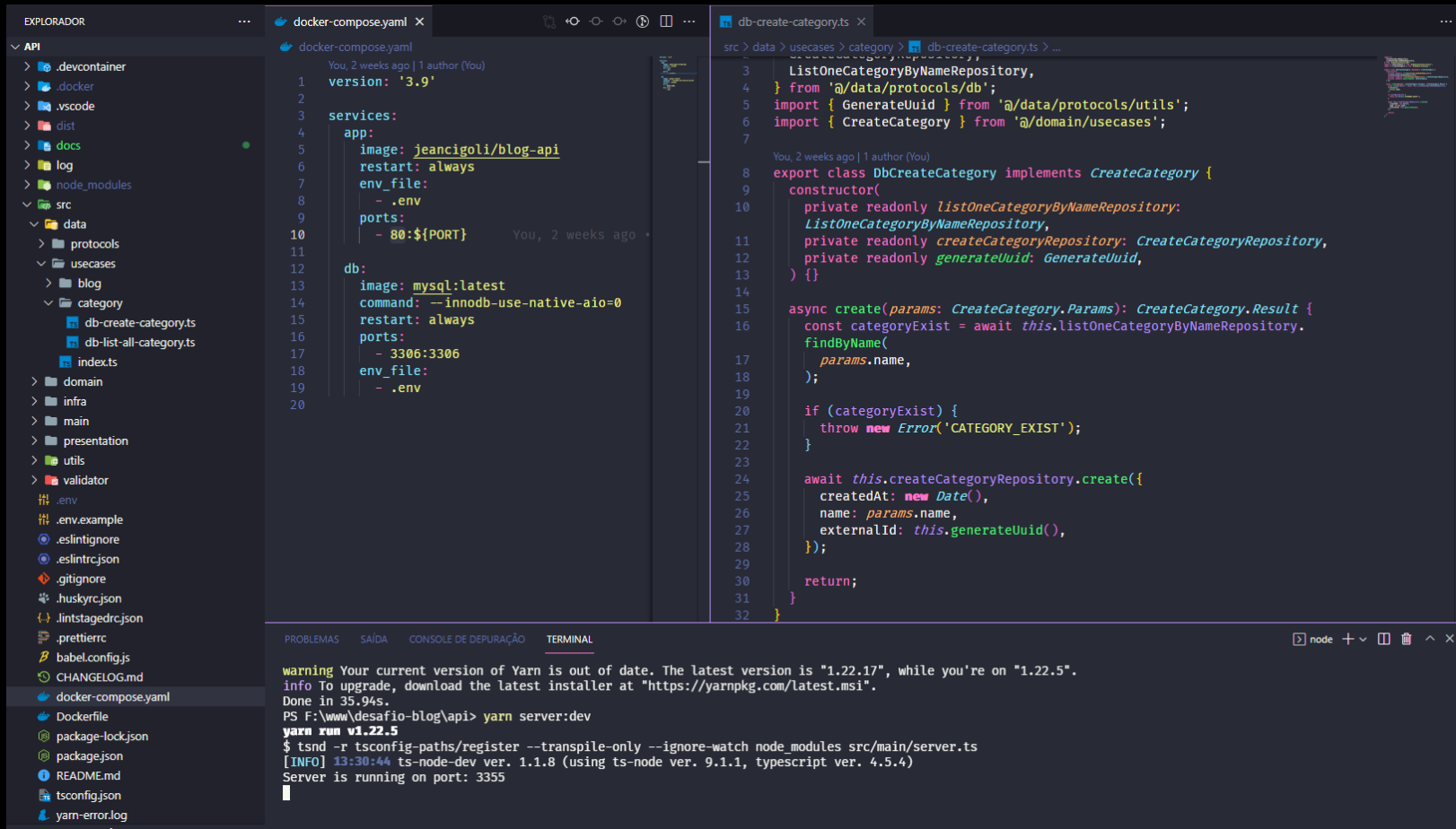
Esse desafio era criar uma API em Node.js para um blog. Esse blog deve conter posts, categorias atreladas e autor.

Para ajudar na utilização da API foi solicitado a documentação, onde optei pelo docusaurus.



A API foi hospedada no EC2 da AWS, e para facilitar o deploy foi gerado uma imagem da aplicação com o Docker.

O seu banco de dados Mysql está no RDS.



The screenshot displays a VS Code editor interface with three main panels:

- EXPLORADOR (Left):** Shows the project file structure. The 'src' directory is expanded, revealing 'data', 'protocols', 'usecases', 'blog', and 'category'. The 'category' directory contains 'db-create-category.ts', 'db-list-all-category.ts', and 'index.ts'.
- docker-compose.yml (Middle):** Displays the configuration for the application and database services.

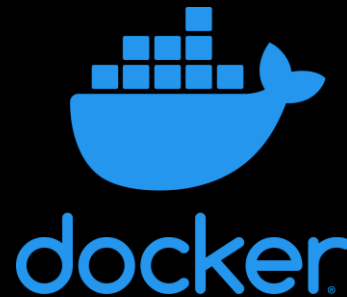
```
1 version: '3.9'
2
3 services:
4   app:
5     image: jeancigoli/blog-api
6     restart: always
7     env_file:
8       - .env
9     ports:
10      - 80:${PORT}
11
12   db:
13     image: mysql:latest
14     command: --innodb-use-native-aio=0
15     restart: always
16     ports:
17      - 3306:3306
18     env_file:
19      - .env
```
- db-create-category.ts (Right):** Shows the TypeScript code for the database service.

```
3 ListOneCategoryByNameRepository,
4 } from '@data/protocols/db';
5 import { GenerateUuid } from '@data/protocols/utills';
6 import { CreateCategory } from '@domain/usecases';
7
8 export class DbCreateCategory implements CreateCategory {
9   constructor(
10     private readonly listOneCategoryByNameRepository:
11       ListOneCategoryByNameRepository,
12     private readonly createCategoryRepository: CreateCategoryRepository,
13     private readonly generateUuid: GenerateUuid,
14   ) {}
15
16   async create(params: CreateCategory.Params): CreateCategory.Result {
17     const categoryExist = await this.listOneCategoryByNameRepository.
18       findByName(
19         params.name,
20       );
21
22     if (categoryExist) {
23       throw new Error('CATEGORY_EXIST');
24     }
25
26     await this.createCategoryRepository.create({
27       createdAt: new Date(),
28       name: params.name,
29       externalId: this.generateUuid(),
30     });
31
32     return;
```
- TERMINAL (Bottom):** Shows the output of the command to start the application.

```
warning Your current version of Yarn is out of date. The latest version is "1.22.17", while you're on "1.22.5".
info To upgrade, download the latest installer at "https://yarnpkg.com/latest.msi".
Done in 35.94s.
PS F:\www\desafio-blog\api> yarn server:dev
yarn run v1.22.5
$ tsnd -r tsconfig-paths/register --transpile-only --ignore-watch node_modules src/main/server.ts
[INFO] 13:30:44 ts-node-dev ver. 1.1.8 (using ts-node ver. 9.1.1, typescript ver. 4.5.4)
Server is running on port: 3355
```

Ferramentas e bibliotecas utilizadas:

- Typescript
- Express
- Git
- Mysql
- Docker
- Docusaurus 2



Para ler mais sobre a parte técnica do projeto ou caso deseje olhar o código, o link do github estará no texto da publicação.

```
{
  "title": "O que é a Tech blog?",
  "short_description": "Conheça a plataforma que te ajuda a entender o mundo tech
de maneira simples e descomplicada.",
  "author": "Jean Cigoli",
  "author_photo": "https://avatars.githubusercontent.com/u/43609958?v=4",
  "categories": [
    "822cb986-d80c-4ad1-8364-9d1444e7cbb6"
  ],
  "sections": [
    {
      "title": "O que é a tech blog?",
      "text": "O desafio que move ....",
      "image": "https://....png"
    }
  ]
}
```



jeancigoli



JeanCigoli