Desafio back-end - Atac Sistemas

Como forma de avaliarmos seus conhecimentos, você deverá nos enviar a resolução do desafio abaixo

Objetivo

A tarefa é construir uma API e banco de dados para a aplicação VUTTR (Very Useful Tools to Remember). A aplicação é um simples repositório para gerenciar ferramentas com seus respectivos nomes, links, descrições e tags.

Utilize um repositório GIT (público) para versionamento e disponibilização de código.

A aplicação deve ser construída utilizando Node + Express + TypeScript + TypeORM + PostgreSQL.

User Stories e wireframes

0): A API deve responder na porta 3000

1): Deve haver uma rota para listar todas as ferramentas cadastradas


```
[
    {
        "id": 1,
        "title": "Notion",
        "link": "https://notion.so",
        "description": "All in one tool to organize teams and ideas. Write, plan, collaborate, and get organized",
        "tags": [
            "organization",
            "planning",
            "collaboration",
            "writing",
            "calendar"
        ]
    },
    {
        "id": 2,
        "title": "json-server",
        "link": "https://github.com/typicode/json-server",
        "description": "Fake REST API based on a json schema. Useful for mocking and creating APIs for front-end devs to consume in
        "tags": [
            "api",
            "json",
            "schema",
            "node",
            "github",
            "rest"
        ]
    },
    {
        "id": 3,
        "title": "fastify",
        "link": "https://www.fastify.io/",
        "description": "Extremely fast and simple, low-overhead web framework for NodeJS. Supports HTTP2.",
        "tags": [
            "web",
            "framework",
            "node",
            "http2",
            "https",
            "localhost"
        ]
    }
]
```

2): Deve ser possível filtrar ferramentas utilizando uma busca por tag

GET /tools?tag=node (node é a tag sendo buscada neste exemplo)


```
[
    {
            "id": 2,
            "title": "json-server",
            "link": "https://github.com/typicode/json-server",
            "description": "Fake REST API based on a json schema. Useful for mocking and creating APIs for front-end devs to consume
            "tags": [
                    "api",
                    "json",
                    "schema",
                    "node",
                    "github",
                    "rest"
            ]
    },
    {
            "id": 3,
            "title": "fastify",
            "link": "https://www.fastify.io/",
            "description": "Extremely fast and simple, low-overhead web framework for NodeJS. Supports HTTP2.",
            "tags": [
                    "web",
                    "framework",
                    "node",
                    "http2",
                    "https",
                    "localhost"
            ]
    }
]
```


3): Deve ser possível filtrar ferramentas utilizando uma busca por título

GET /tools?title=fastify (fastify é o título da ferramenta que está sendo buscada neste exemplo)


```
[
    {
            "id": 3,
            "title": "fastify",
            "link": "https://www.fastify.io/",
            "description": "Extremely fast and simple, low-overhead web framework for NodeJS. Supports HTTP2.",
            "tags": [
                    "web",
                    "framework",
                    "node",
                    "http2",
                    "https",
                    "localhost"
            ]
    }
]
```


4): Deve haver uma rota para cadastrar uma nova ferramenta

O corpo da requisição deve conter as informações da ferramenta a ser cadastrada, sem o ID (gerado automaticamente pelo servidor). A resposta, em caso de sucesso, deve ser o mesmo objeto, com seu novo ID gerado.

POST /tools Content-Type: application/json

```
{
    "title": "hotel",
    "link": "https://github.com/typicode/hotel",
    "description": "Local app manager. Start apps within your browser, developer tool with local .localhost domain and https out of
    "tags":["node", "organizing", "webapps", "domain", "developer", "https", "proxy"]
}
```

Resposta:

Status: 201 Created

Content-Type: application/json

```
{
        "id":5,
        "title": "hotel",
        "link": "https://github.com/typicode/hotel",
        "description": "Local app manager. Start apps within your browser, developer tool with local .localhost domain and https out
        "tags":["node", "organizing", "webapps", "domain", "developer", "https", "proxy"]
}
```

5): O usuário deve poder remover uma ferramenta por ID

DELETE /tools/:id

Resposta:

Status: 204 No Content

O que será avaliado

Queremos avaliar sua capacidade de desenvolver e documentar um back-end para uma aplicação. Serão avaliados:

- Código bem escrito e limpo;
- Seu conhecimento em banco de dados, requisições HTTP, APIs REST, etc;
- Sua capacidade de se comprometer com o que foi fornecido;

O mínimo necessário

- Uma aplicação contendo uma API real simples, sem autenticação, que atenda os requisitos descritos acima, fazendo requisições à um banco de dados para persistência;
- README.md contendo informações básicas do projeto e como executá-lo;

Diferenciais

- Documentação dos seus endpoints (ex: Arquivo texto / Swagger / Postman)
- Migrations para configuração do banco de dados utilizado;
- Cobertura de testes;
- Autenticação e autorização (JWT);
- Disponibilizar o código rodando em algum ambiente cloud (AWS, Heroku, DigitalOcean);
- Utilizar ferramentas para análise estática de código (Prettier, ESLint);