

DESAFIO BACK-END

O desafio consiste em desenvolver uma solução que permita interagir e manipular dados de utilizadores do GitHub. Para isso, deverá ser implementada uma API que disponibiliza os seguintes endpoints:

Endpoint 1: Deverá permitir copiar para uma base de dados local, todos os repositórios públicos de um determinado utilizador do GitHub. Para isso, o endpoint deve receber por parâmetro o nome do utilizador desejado. Os seguintes dados precisam ser armazenados:

- ID do repositório;
- Nome do repositório;
- Descrição;
- URL;
- Linguagem principal;
- Data de criação do repositório;
- ID do utilizador;
- Login do utilizador;
- Avatar do utilizador.

Endpoint 2: Deverá receber o nome de um determinado utilizador e listar todos os seus repositórios, que estão armazenados na base de dados local. Os seguintes dados precisam ser retornados:

- ID do repositório;
- Nome do repositório;
- Descrição;
- URL;
- Linguagem principal;
- Data de criação do repositório.

Endpoint 3: Deverá realizar uma procura por repositórios armazenados na base de dados. Para isso receberá por parâmetro as palavras a serem pesquisadas nos dados dos repositórios.

Endpoint 4: Retornar estatísticas calculadas apenas com os dados locais já sincronizados (via Endpoint 1):

- Parâmetros (query):
 - **user** (opcional): login do usuário
 - **Ausente** ⇒ estatísticas globais (todos os utilizadores salvos)
 - **Presente** ⇒ estatísticas somente desse usuário
 - **topN** (opcional, default: 5, máx: 20): tamanho dos rankings;
- Exemplo de retorno:
 - **summary:**
 - total_repos;
 - total_users → aparece apenas para estatísticas global
 - **languages:** linguagens por quantidade de repositórios
 - **top_users_by_repos:**
 - Somente no global: utilizadores com mais repositórios (top N)
 - **timeline_created_monthly:**

- Histograma de repositórios criados por mês

GITHUB API'S:

Repositórios: *GET* <https://api.github.com/users/{user}/repos>

Users: *GET* <https://api.github.com/users/{user}>

OBSERVAÇÕES:

A solução deve ser desenvolvida preferencialmente em Node, com TypeScript e NestJS (ou framework similar).

Será um adicional se a solução contiver configurações de Docker ou Docker Compose, que permitam executar a aplicação.