

DESAFIO

Você deve criar uma aplicação que realize um processo de *ETL* (Extract, Transform and Load). Os passos do processo estão detalhados abaixo.

1. Extract

Realize chamadas na nossa API REST para extrair um conjunto de números da nossa base de dados.

Exemplo:

GET <http://challenge.dienekes.com.br/api/numbers?page=1>

Retorno:

```
{ "numbers": [ 0.4971795774527892, 0.7311238428477732, 0.004048275097350857 ] }
```

Note que a nossa API recebe o parâmetro "page" na url. Você deve extrair os números de todas as páginas disponíveis na nossa base de dados. Você vai saber que conseguiu extrair todos os números disponíveis quando solicitar uma página e o nosso servidor retornar um array vazio.

Exemplo de retorno quando não existir mais números a serem extraídos:

```
{ "numbers": [] }
```

2. Transform

A etapa de transformação consiste em ordenar todos os números extraídos na etapa anterior.

IMPORTANTE: a ordenação deve ser feita com o conjunto final contendo todos os números extraídos de todas as páginas.

IMPORTANTE 2: Você deve implementar o algoritmo de ordenação. Não é permitido utilizar nenhum recurso da linguagem que faça toda a ordenação para você.

3. Load

A aplicação deve expor uma API que disponibiliza o conjunto final de números ordenados da etapa de *transform*. Fique a vontade para escolher o tipo da API (rest, soap, graphql etc), modelagem dos métodos e formato dos dados.

Restrições

- Você deve implementar o algoritmo de ordenação da etapa de "Transform". Não é permitido utilizar nenhuma função/método que faça toda a ordenação pra você.
- Não é permitido copiar a solução de outra pessoa. Integridade é um dos valores da Cross Commerce.

Fora essas duas restrições, você está livre para escolher a linguagem, modelagem, framework etc.

O que vamos avaliar

- Corretude;
- Legibilidade;
- Tolerância a falha;
- Complexidade algorítmica;
- Testes automáticos.