

## DESAFIO/CASE:

### Descrição:

Considere o seguinte cenário para implementação de uma nova API. Esta implementação será para criação, alteração e consulta de usuários:

### A sua api deverá validar pré-requisitos de senha, nos seguintes formatos:

- Nove ou mais caracteres
- Ao menos 1 dígito
- Ao menos 1 letra minúscula
- Ao menos 1 letra maiúscula
- Ao menos 1 caractere especial (!@#\$%^&\*()\_+)
- Não repetir caracteres dentro do mesmo conjunto

### Exemplo String inválidas (false) e válidas (true):

- "" = false
- "aa" = false
- "ab" = false
- "aaabbbCc" = false
- "EtPd8!goo" = false
- "EtPd8 goo" = false
- "EtPd8!gof" = true

### Trabalho:

Construa um backend em (GO ou Java ou NodeJs) e exponha um API Rest com os seguintes inputs/outputs:

#### Criação de Usuário

- Input: nome, password e email

#### Consulta de Usuário

- Input: identificador único
- Output: nome, password e email

#### Listagem de Usuários

- Output: nome, password e email

#### Alteração de Usuário

- Input: nome, password e email
- Output: nome, password e email

\* Você poderá utilizar um banco de dados a sua escolha (InMemory, SQL ou NoSQL).

\* As mensagens deverão ser no formato JSON.

### Expectativa que iremos analisar:

- Testes Unitários
- Desenho de API (OAS)

- Clean Code
- Twelve factory
- Doc. Readme.md
- Tratamento de erros
- Estrutura do projeto

**Entregáveis:**

- Código no github do candidato
- Documentação Readme
- Instruções de como rodar a api

**Plus:**

- Dockerfile/Docker compose
- Collection Postman
- Autenticação OAuth2.0
- API rodando em alguma cloud
- Filtros na consulta
- Teste de Carga(Load test) + Resultados.