



CUCOHealth | Challenge | Full Stack

## INTRODUÇÃO

Bem-vindo(a). Este é o desafio Full Stack!

O objetivo deste desafio é avaliar suas habilidades de programação.

Quando sua solução estiver pronta, basta enviar um e-mail para [developer@cucohealth.com](mailto:developer@cucohealth.com) com o link do seu repo no Github (ou similar), junto com instruções de como rodar o projeto. Em seguida, enviaremos o feedback e as instruções dos próximos passos!

## CRITÉRIOS DE AVALIAÇÃO

Além dos requisitos descritos no teste, iremos olhar para os seguintes critérios durante a correção do desafio:

- Padrões de código;
- Boas Práticas;
- Organização/Estrutura do projeto
- Padrão de commits;
- Funcionamento das funcionalidades.

## SERIA LEGAL

- Documentação;
- Testes unitários;
- Fazer deploy no heroku, vercel ou em outro local de sua preferência e disponibilizar um link de visualização.



## BACK-END

### CONTEXTO

Crie uma API HTTP responsável pelo gerenciamento de *clientes*.

O nosso cliente precisa conter os seguintes campos:

- Nome completo;
- CPF;
- Data de nascimento;
- Telefone (opcional).

Esta API DEVE conter endpoints para as seguintes ações:

- Criar um cliente;
- Deletar um cliente;
- Listar clientes criados;
- Buscar clientes por nome e/ou CPF.

### INSTRUÇÕES

Chegou a hora de colocar a mão na massa!

- Utilize PHP, Java ou C#;
- O retorno da API deve ser em JSON;
- O uso de framework é permitido e você poderá escolher a de sua preferência;
- Você tem total liberdade para escolher a forma que esses dados serão gravados.
- Será considerado com diferencial alguma forma de documentação, seja collection de Postman (ou algum outro programa), swagger ou alguma outra forma de documentação que nos auxilie.



## FRONT-END

### CONTEXTO

Crie um front end para realizar o gerenciamento de *clientes*, consumindo a API criada anteriormente.

Para este gerenciamento precisamos ter pelo menos as seguintes telas:

- Listagem de Clientes - elementos importantes:
  - Listagem de clientes;
  - Opção de deletar um cliente;
  - Botão de cadastro de clientes;
  - Input para busca de cliente por nome e/ou CPF;
  - Atualização da lista após a busca.
- Cadastro de Cliente - elementos importantes:
  - Input para nome;
  - Input para CPF;
  - Input para data de nascimento;
  - Input para telefone (opcional).

### INSTRUÇÕES

Chegou a hora de colocar a mão na massa!

- Utilize Angular, React ou Vue.js;
- Você tem total liberdade para montar o layout e organizar os componentes na tela conforme sua preferência.