

## Teste de Front-end

---

Este teste é apresentado aos candidatos as vagas de desenvolvimento Front-end para avaliar os quesitos técnicos.

### O Desafio

Seu objetivo é criar um simples app que deve conter duas páginas, uma que exibe um formulário com os campos abaixo, e outra que liste os dados cadastrados.

- Nome completo
- CPF
- Telefone
- Email

### Pré-requisitos:

- Deve ser possível criar, listar e excluir os dados cadastrados pelo formulário;
- Os inputs de texto e botão devem ter a aparência conforme o guia de estilo abaixo (com validações);
- Fazer a persistência dos dados no `localStorage` ou `IndexedDB`;
- Recomendado uso de ECMAScript 6+ e React.JS;

Para ter o estado inicial da lista de usuário utilizar este recurso abaixo:

GET <https://private-9d65b3-tinnova.apiary-mock.com/users>

Response:

```
[
  {
    "name": "Joao da Silva",
    "cpf": "26899337649",
    "phone": "4233335555",
    "email": "joao@joaosilva.com.br"
  },
  {
    "name": "Maria Antonieta",
    "cpf": "65138896180",
    "phone": "1255553333",
    "email": "maria@mariaantonieta.com.br"
  },
  {
    "name": "Luiz Souza",
    "cpf": "32420496329",
    "phone": "1144446666",
```

```
"email": "luiz@luizsouza.com.br"
}
]
```

A partir deste ponto utilizar o `localStorage/IndexedDB` para persistir localmente as informações.

Save:

```
{
  "name": "Marcos Silva",
  "cpf": "24146898030",
  "phone": "1188887777",
  "email": "marcos@marcoossilva.com.br"
}
```

Lista local:

```
[
  {
    "name": "Joao da Silva",
    "cpf": "26899337649",
    "phone": "4233335555",
    "email": "joao@joaosilva.com.br"
  },
  {
    "name": "Maria Antonieta",
    "cpf": "65138896180",
    "phone": "1255553333",
    "email": "maria@mariaantonieta.com.br"
  },
  {
    "name": "Luiz Souza",
    "cpf": "32420496329",
    "phone": "1144446666",
    "email": "luiz@luizsouza.com.br"
  },
  {
    "name": "Marcos Silva",
    "cpf": "24146898030",
    "phone": "1188887777",
    "email": "marcos@marcoossilva.com.br"
  }
]
```

Plus:

- A página ser responsiva;
- Permitir edição;
- Uso de pré-processador css;
- Testes End to End;

## O que esperamos:

- Testes, no mínimo testes unitários;
- Padrão de Projeto e boas práticas de Orientação a Objetos;
- Utilizar ECMAScript 6+;
- Criar um passo a passo de como rodar sua aplicação ([Sugestão](#));
- Criar uma breve descrição da solução utilizada.

## Guia de estilo

---

### Input:

- Cor da fonte sem foco: #efeed.
- Cor da fonte com foco: #333333.
- Cor da borda: #efeed.

Nome completo (sem abreviações)

---

### Input Inválido:

- Cor da fonte: #eb4a46.
- Cor da borda: #eb4a46.

Nome completo (sem abreviações)

---

Campo deve conter 3 caracteres ou mais

### Botão Habilitado:

- Cor da fonte com foco: #ffffff.
- Cor de background: #00c8b3.

Cadastrar

Botão Hover:

- Opacidade do botão com hover: 70%.

Cadastrar

Botão Desabilitado:

- Cor da fonte sem foco: #dddcdd.
- Cor de background: #f6f6f6.

Cadastrar

Botão Loading:



*\*Criar animação de loading ao clicar no submit*

## Exemplo Final

Nome completo (sem abreviações)

**José Santos**

E-mail

**josesantos@gmail.com**

CPF

**413.871.851-60**

Telefone

**(42) 99999-9999**

**Cadastrar**

**\*\* Use sua criatividade para criar a página para listar os dados cadastrados \*\***