

Prezado Candidato, boa tarde!

Informo que estamos evoluindo para a fase de teste técnico neste processo seletivo, por favor, poderia providenciar o envio do teste conforme abaixo:

Vaga: Desenvolvedor Front-End Angular

Sobre o Centro de Pesquisas Avançadas Wernher von Braun:

O **Centro de Pesquisas Avançadas Wernher von Braun** é uma instituição que atua em P&D (Pesquisa e Desenvolvimento) e Inovação para Empresas e Instituições em Geral.

Foi criado em 1997 e, desde o início de suas atividades, trabalha desenvolvendo soluções tecnológicas nas áreas de Software (de Sistemas de TI a Embarcados passando por Aplicativos) e Hardware (Componentes Semicondutores e Equipamentos) em soluções de inovação para os mercados de *Intelligent Transportation Systems* (ITS), Automação e Inteligência de Varejo, Sistemas Automotivos, Big Data, Ferramentas para Estruturação de Serviços, Aplicativos para Mídias de Consumo, entre outros.

Descrição da vaga:

- Atuar com desenvolvimento de sistemas (Web e Mobile) em projetos de pesquisa, aplicações industriais e de logística entre outros;

Desafio de Desenvolvimento de Front-End:

Objetivo:

Criar uma interface de usuário para utilização em uma aplicação de cadastro e compartilhamento de acesso à dispositivos de IoT

Instruções:

Contexto: Você está trabalhando em um projeto colaborativo para cadastro e compartilhamento de dados de dispositivos de IoT. Nesse projeto, proprietários de dispositivos IoT podem fazer o cadastro nesses dispositivos na plataforma, incluindo informações de acesso, comandos disponibilizados pelos dispositivos e dados retornados por estes comandos. Nesse projeto, você deve desenvolver a aplicação em Angular, acessando a API apresentada no anexo deste desafio, para realizar a gestão de dispositivos (cadastro, consulta, atualização e remoção - CRUD)

Requisitos Funcionais:

1. Os usuários devem poder autenticar-se na aplicação para acessar as funcionalidades de gestão de dispositivos, a API ficará responsável por realizar a associação de um dispositivo ao usuário logado, criando o conceito de “proprietário”.
2. Os usuários devem ser capazes de cadastrar novos dispositivos IoT na plataforma

3. Deve ser possível visualizar uma lista de todos os dispositivos cadastrados na plataforma, incluindo a descrição e o fabricante do dispositivo.
4. Os usuários proprietários devem poder editar as informações dos dispositivos cadastrados, como nome, descrição, informações de acesso e lista de comandos.
5. Deve ser possível aos usuários proprietários excluir dispositivos da plataforma
6. A aplicação deve ser responsiva e funcionar adequadamente em diferentes dispositivos e tamanhos de tela, garantindo uma experiência consistente para os usuários em desktops, tablets e smartphones.

Requisitos de Design:

1. A interface do usuário deve ser intuitiva e fácil de usar, com navegação clara e fluxos de trabalho lógicos para as funcionalidades de gestão de dispositivos.
2. Utilize um design limpo e moderno, com uso adequado de cores, tipografia e espaçamento para criar uma experiência visualmente atraente e profissional.
3. Forneça feedback visual claro para as ações dos usuários, como confirmação de sucesso após o cadastro ou atualização de um dispositivo, ou mensagens de erro em caso de falha.
4. Utilize adequadamente os princípios de gerenciamento de estado no Angular para manter a consistência e a integridade dos dados da aplicação.
5. Implemente medidas de segurança adequadas para proteger os dados dos usuários e garantir a integridade das operações de gestão de dispositivos.

Entregáveis:

1. Código Fonte: Repositório no Github do candidato, contendo o código fonte do projeto criado
2. Explicação de Decisões de Design e Implementação: O arquivo README do repositório Git deve detalhar, de forma clara, as decisões de design e implementação adotadas no projeto, bem como sugestões de melhorias e avanços futuros. É importante o candidato organizar adequadamente essas informações para que seja fácil identificá-las e avaliá-las

Prazo de Entrega: 7 dias a partir da data de recebimento do desafio.

Os entregáveis devem ser enviados para selecao@vonbraunlabs.com.br com o título “Desafio Front-End Angular”

Anexo: Documentação da API

```
Unset
openapi: 3.0.0
info:
  title: Community IoT Device (CIoTD)
  description: >-
    <p>A CIoTD é uma plataforma colaborativa para compartilhamento de acesso à dados de dispositivos IoT.</p>
    <p>Através dela, colaboradores podem cadastrar seus dispositivos, permitindo que qualquer pessoa possa coletar os dados desses dispositivos e utilizar em suas aplicações.</p>
  version: 1.0.0
  contact: {}
tags:
  - name: Devices
    description: Provê operações para cadastramento e gerenciamento de dispositivos
paths:
  /device:
    get:
      summary: >-
        Retorna uma lista contendo os identificadores dos dispositivos cadastrados na plataforma
      tags:
        - Devices
      responses:
        '200':
          description: >-
            Requisição executada com sucesso
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/DeviceList'
        '401':
          description: >-
            As credenciais fornecidas pelo usuário são inexistentes ou inválidas
    post:
      summary: >-
        Cadastra um novo dispositivo na plataforma
      tags:
        - Devices
      requestBody:
        description: Detalhes do dispositivo sendo cadastrados
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Device'
      responses:
        '201':
```

```
    description: >-
      Requisição realizada com sucesso
    headers:
      Location:
        description: URL de acesso aos dados dispositivo recém cadastrado
        schema:
          type: string
  /device/{id}:
    get:
      summary: >-
        Retorna os detalhes de um dispositivo
      tags:
        - Devices
      parameters:
        - name: id
          in: path
          description: Identificador do dispositivo para o qual os detalhes devem ser
retornados
        required: true
        schema:
          type: string
          format: byte
      responses:
        '200':
          description: >-
            Requisição realizada com sucesso
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Device'
        '404':
          description: Dispositivo não encontrado
    put:
      summary: >-
        Atualiza os dados de um dispositivo
      tags:
        - Devices
      parameters:
        - name: id
          in: path
          description: Identificador do dispositivo para o qual os detalhes devem ser
atualizados
        required: true
        schema:
          type: string
          format: byte
      responses:
        '200':
```

```
description: >-
  Requisição realizada com sucesso
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Device'
'401':
  description: A solicitação não foi realizada pelo proprietário do dispositivo
'404':
  description: Dispositivo não encontrado
delete:
  summary: >-
    Remove os detalhes de um dispositivo
  tags:
    - Devices
  parameters:
    - name: id
      in: path
      description: Identificador do dispositivo para o qual os detalhes devem ser
removidos
    required: true
    schema:
      type: string
      format: byte
  responses:
    '200':
      description: >-
        Requisição realizada com sucesso
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Device'
    '401':
      description: A solicitação não foi realizada pelo proprietário do dispositivo
    '404':
      description: Dispositivo não encontrado
security:
  - basicAuth: []
components:
  securitySchemes:
    basicAuth:
      type: http
      scheme: basic
schemas:
  Command:
    type: object
  properties:
    command:
```

```

    type: string
    description: Sequencia de bytes enviados para execução do comando
  parameters:
    type: array
    description: Lista de parâmetros aceitas pelo comando
    items:
      $ref: '#/components/schemas/Parameter'
  CommandDescription:
    type: object
    properties:
      operation:
        type: string
        description: Nome da operação executada pelo dispositivo
      description:
        type: string
        description: Descrição e detalhes adicionais sobre a operação e/ou o comando
      command:
        $ref: '#/components/schemas/Command'
      result:
        type: string
        description: Descrição do resultado esperado da execução do comando
      format:
        type: string
        description: Definição, usando o padrão OpenAPI para especificação de schemas
de dados, do formato dos dados retornados pelo comando.
  Device:
    type: object
    properties:
      identifier:
        type: string
        description: Identificador do dispositivo
      description:
        type: string
        description: Descrição do dispositivo, incluindo detalhes do seu uso e das
informações geradas
      manufacturer:
        type: string
        description: Nome do fabricante do dispositivo
      url:
        type: string
        description: URL de acesso ao dispositivo
      commands:
        type: array
        description: Lista de comandos disponibilizada pelo dispositivo
        items:
          $ref: '#/components/schemas/CommandDescription'
  DeviceList:
    type: array

```

```
items:
  type: string
  description: >-
    Identificador do dispositivo
Parameter:
  type: object
  properties:
    name:
      type: string
      description: nnome do parâmetro
    description:
      type: string
      description: Descrição do parâmetro, incluindo detalhes de sua utilização,
valores possíveis e funcionamento experado da operação de acordo com esses valores
```