

Vaga: Desenvolvedor Backend

Prezado(a) Candidata, segue este anexo com o Desafio e a API para realizarem o Teste, conforme descrito abaixo:

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 (APIs e Web) em projetos de pesquisa, aplicações industriais e de logística entre outros;

Desafio de Desenvolvimento BackEnd:

Objetivo:

Realizar a integração de sistemas, através de uma plataforma colaborativa de cadastro de dispositivos IoT, para fornecimento de *insights* para tomada de decisão de plantio

Instruções:

Contexto: Você está trabalhando em um projeto de auxílio à tomada de decisão para uma indústria agrária. Essa indústria utiliza equipamentos da PredictWeather medir a densidade de chuva. A PredictWeather possui vários sensores, mas, o equipamento que mede a densidade de chuva opera com o comando ***get_rainfall_intensity***. A API apresentada no anexo deste desafio destina-se ao cadastro e compartilhamento de dados de dispositivos IoT. Considerando o anteposto, desenvolva uma aplicação que, utilizando a API apresentada no anexo, retorne uma lista da volumetria instantânea de chuva medidas por todos os sensores cadastrados na plataforma com essa capacidade.

Requisitos Funcionais:

1. Os usuários devem poder autenticar-se na aplicação para pode consultar os dados de volumetria de chuva.
2. Apenas dispositivos cujo fabricante seja a PredictWeater e que possuem o comando ***get_rainfall_intensity*** devem ser considerados pela aplicação
3. A aplicação deve ter um endpoint que retorne uma lista dos dispositivos cadastrados na API apresentada no anexo que atendam aos requisitos definidos no item 2.
4. Para cada dispositivo na lista definida no item 4, a aplicação deve conectar-se à esse dispositivos e, através do protocolo telnet, e enviar o comando apropriado, para receber as medições

5. Os dados das medições devem ser incluídos na mensagem de resposta do endpoint definido no item 3.
6. A aplicação devem aplicar métodos para otimizar as requisições e reduzir o tempo de resposta.

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 Back-End”

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
```