

---

# Desafio Técnico

## 1. INTRODUÇÃO

O time de controle da companhia **Pedra Pagamentos** é responsável por efetuar o acompanhamento de todos os KPIs e OPIs, de modo a garantir que as metas estabelecidas serão atingidas. A companhia ainda é uma Startup, e muitos processos precisam ser amadurecidos, pois são manuais e demorados.

Você é responsável por automatizar o processo de cálculo de alguns indicadores da operação. Se for bem sucedido, os analistas poderão investir mais tempo na análise de dados (hoje esse tempo é desperdiçado com extração e tratamento de dados para que sejam posteriormente apurados).

Neste primeiro momento o foco será a automatização dos indicadores da operação de [Last Mile](#). São eles:

- Produtividade por Green Angel
- SLA de cada base logística
- SLA de cada Green Angel



---

## 2. INSTRUÇÕES

Você recebeu um arquivo CSV com todos os atendimentos efetuados pelo time de logística nos últimos 12 meses.

Agora sua missão é:

1. Desenvolver uma API REST que tenha os seguintes métodos implementados:
  - a. Inserir um novo atendimento
  - b. Consultar atendimentos (com possibilidade de filtrar por campos que você considere relevantes)
  - c. Atualizar dados de um atendimento

Essa API REST deve ser desenvolvida com a linguagem Python 3. Caso opte pela utilização de um framework, este deve estar entre as seguintes opções: *Flask* ou *Falcon*.

2. Persistir os dados referentes a cada atendimento em um banco de dados PostgreSQL.
3. Criar um script para efetuar a carga do arquivo CSV no banco recém criado. Considere que a partir da data de hoje todos os novos atendimentos efetuados serão registrados através da API que você desenvolveu.

### 2.1 Requisitos

- Utilize container Docker para executar a API e o banco de dados.
- Utilize a linguagem Python 3, com Flask ou Falcon para criação da API
- Banco de dados PostgreSQL
- Projeto entregue pelo github
- README com instruções para instalação e execução do projeto

### 2.2 Bônus

- Docker-compose para que todos os containers possam ser inicializados em conjunto
- Testes unitários e end-to-end
- Deploy via CI/CD numa cloud (preferencialmente GCP)
- Autenticação da api
- Criar monitoramentos como tempo de resposta, latência, etc. em requisições testes realizadas
- Seguir Clean Architecture ou Alguma metodologia de Design / Arquitetura de Software
- Padronização de commits

---

## 2.3 Critérios de Avaliação

- Princípios SOLID
- Documentação
- Padrões de Projeto
- Legibilidade e organização do código
- Capacidade de abstração do problema
- Tratamento de Erros
- Criatividade

## 3. GLOSSÁRIO

- **SLA:** Quantidade de atendimentos no prazo / Quantidade total de atendimentos
- **Produtividade:** Quantidade de atendimentos / dia útil
- **Green Angel:** Técnico de logística

## 4. CONTATO

Em caso de dúvidas, enviar e-mail para **[sistemas@stone.com.br](mailto:sistemas@stone.com.br)**