



# Teste Desenvolvedor Backend Python

Teste de Habilidades Específicas

*A Greenole é mais do que apenas uma plataforma de análise de dados; é um catalisador para mudanças sustentáveis e eficientes em qualquer empresa. Nosso foco é tornar a sustentabilidade não apenas uma meta, mas uma realidade alcançável para as empresas de todos os tamanhos.*

*Nossa plataforma utiliza tecnologia de ponta para coletar, analisar e interpretar dados, oferecendo insights valiosos que podem transformar operações comerciais. Os benefícios vão além do simples corte de custos ou aumento da eficiência. A Greenole está comprometida em ajudar as empresas a fazer escolhas mais responsáveis que beneficiam não apenas seus resultados financeiros, mas também o meio ambiente.*

*Com Greenole, as empresas obtêm uma visão mais clara de como suas operações afetam o mundo ao seu redor e, mais importante, como podem melhorá-lo. Nossas soluções são projetadas para serem escaláveis e adaptáveis, atendendo às necessidades únicas de cada cliente.*

*Em resumo, a Greenole oferece uma nova perspectiva sobre a sustentabilidade e eficiência operacional, impulsionada por análises e insights. Nós fornecemos as ferramentas, você fornece a ação.*

## Instruções:

A resolução pode ser enviada através de um arquivo zip com os arquivos da solução ou em um repositório público no GitHub ou em um repositório privado cujo usuário "rodrigobraga" seja adicionado de leitura.

Não existe exatamente um prazo para entrega, mas entendemos como relevante que ele seja completado em até uma semana.

Desenvolva uma aplicação web usando o framework Django, com ao menos dois modelos e que atenda aos seguintes requisitos:

### 1. **\*\*Arquitetura\*\***.

Desenvolva uma API que simule uma ferramenta de exibição de dados vindo de sensores IoT que leem a temperatura de determinado local.

Primeiro simule o envio de dados vindo de sensores para 3 clientes diferentes. (Cada cliente contendo 2 sensores por exemplo) Esses dados devem ir para uma aplicação web que utiliza um broker mqtt para distribuí-los e salvar no banco de dados.

Segundo desenvolva uma API rest que faça a leitura desses dados:

- Faça endpoints para o CRUD para Clientes, Sensores e Usuários.
- Um endpoint que deve retornar o último dado do sensor.
- Um endpoint que deve retornar a média de temperatura por dia.
- Um endpoint que retorna os dados de 100 em 100, paginados e ordenados pela ordem de envio mais recente.

Alguns pontos de validação:

- Somente usuários logados no sistema podem usar esses endpoints acima.
- Um usuário pode somente ver os dados do cliente ao qual ele pertence.
- Somente um usuário Admin pode adicionar Clientes.

### 2. **\*\*Testes\*\***:

- Escreva testes de unidade abrangentes para as principais funcionalidades da aplicação.
- Crie testes de integração para garantir que os componentes da aplicação funcionem bem em conjunto.
- Utilize bibliotecas de teste do Django, como `unittest` ou `pytest`.

### 3. **\*\*Bancos Relacionais e Não-Relacionais\*\***:

- Configure um banco de dados relacional (PostgreSQL) para armazenar informações essenciais da aplicação.
- Integre um banco de dados não relacional (por exemplo, MongoDB) para armazenar dados que se beneficiam de um modelo de dados flexível.



# Greenole

## 4. **\*\*Cache\*\***:

- Implemente cache em partes críticas da aplicação para melhorar o desempenho.
- Utilize a funcionalidade de cache incorporado no Django ou uma solução externa, usando o Redis ou Memcached.

## 5. **\*\*Segurança Básica\*\***:

- Implemente medidas de segurança básicas, como autenticação e autorização para proteger os recursos da API.
- Valide e sanitize as entradas dos usuários para prevenir ataques de injeção.

## 6. **\*\*Alto Desempenho\*\***:

- Otimize consultas de banco de dados para garantir um alto desempenho.
- Utilize ferramentas de monitoramento de desempenho para identificar gargalos e otimizar a aplicação.