

# Case Técnico: Integração com HubSpot

## Objetivo

Este desafio tem como objetivo avaliar sua capacidade de compreensão de um problema técnico, análise de soluções, estruturação do raciocínio lógico e aplicação de boas práticas de código, segurança e.

Você deverá desenvolver uma **API REST em Java** para integrar com a **API do HubSpot**, implementando autenticação via **OAuth 2.0**, mais especificamente com o fluxo de **authorization code flow**, a implementação de endpoint de integração com a API e o recebimento de notificações via **webhooks**.

---

## Descrição do Desafio

Sua tarefa é criar uma aplicação backend com os seguintes **endpoints obrigatórios**:

1. **Geração da Authorization URL:**
    - Endpoint responsável por gerar e retornar a URL de autorização para iniciar o fluxo OAuth com o HubSpot.
  2. **Processamento do Callback OAuth:**
    - Endpoint recebe o código de autorização fornecido pelo HubSpot e realiza a troca pelo **token de acesso**.
  3. **Criação de Contatos:**
    - Endpoint que faz a criação de um Contato no CRM através da API. O endpoint deve respeitar as políticas de rate limit definidas pela API.
  4. **Recebimento de Webhook para Criação de Contatos:**
    - Endpoint que escuta e processa eventos do tipo **"contact.creation"**, enviados pelo webhook do HubSpot.
-

## Requisitos Técnicos

- Desenvolver a API REST em **Java** utilizando o framework **Spring Boot ou Play Framework**.
  - Implementar **boas práticas de segurança** conforme recomendações da **documentação do HubSpot**.
  - Seguir **boas práticas de código**, incluindo **separação de responsabilidades, tratamento adequado de erros e estruturação clara do código**.
  - Disponibilizar **instruções detalhadas** de como executar a aplicação.
- 

## Recursos Úteis

- Criar uma conta de desenvolvedor no HubSpot: [HubSpot Developer Account](#)
  - Guia rápido sobre OAuth no HubSpot: [OAuth Quickstart](#)
- 

## O que será avaliado?

- **Clareza e organização do código**  
Sua capacidade de escrever um código limpo, legível e bem estruturado, facilitando a manutenção e entendimento por pessoas de diferentes níveis de experiência.
  - **Aderência às boas práticas**  
Esperamos que sejam aplicadas boas práticas tanto de codificação (princípios como SOLID, KISS e Clean Code) quanto de segurança, especialmente seguindo as recomendações do HubSpot e as convenções do protocolo de autenticação.
  - **Equilíbrio entre qualidade e pragmatismo**  
Valorizamos soluções eficientes, adequadas ao contexto e que não incorporem complexidade desnecessária (*overengineering*). A solução deve ser suficientemente robusta e bem elaborada, mas sem sacrificar a simplicidade e produtividade.
  - **Domínio da linguagem e framework**  
A solução deve demonstrar seu conhecimento e habilidade na utilização da linguagem Java e do framework escolhido, fazendo uso apropriado de recursos nativos e boas práticas do ecossistema.
-

# Entrega

O prazo para a entrega do desafio é de **cinco dias**. Você deverá disponibilizar:

1. O **código-fonte** em um repositório **GitHub**.
2. Um arquivo **README.md** contendo **instruções detalhadas** para execução do projeto.
3. Uma breve **documentação técnica**, explicando as decisões tomadas, motivação para uso de libs e possíveis melhorias futuras.
4. Você deve enviar o link do projeto no repositório em um e-mail para os seguintes destinatários:
  - a. [thais.dias@meetime.com.br](mailto:thais.dias@meetime.com.br)
  - b. [joao@meetime.com.br](mailto:joao@meetime.com.br)
  - c. [william.willers@meetime.com.br](mailto:william.willers@meetime.com.br)
  - d. [victor@meetime.com.br](mailto:victor@meetime.com.br)

---

💡 **Dica:** Fique a vontade para utilizar libs na sua implementação, mas não esqueça de descrever a decisão de incluí-la no arquivo **README.md**.

Boa sorte! 🚀