

Documentação Técnica

Visão Geral

Esta aplicação tem como objetivo integrar-se à API da HubSpot utilizando autenticação OAuth2, permitindo a criação e listagem de contatos da plataforma. A aplicação foi construída com foco em boas práticas de arquitetura (SOLID, Clean Architecture) e resiliência.

Decisões de Projeto

Arquitetura em Camadas com Separação de Responsabilidades

- **Camada de Controller:** expõe os endpoints REST.
- **Camada de UseCase/Service:** concentra a lógica de negócio e orquestra as ações.
- **Ports and Adapters (Hexagonal Architecture):** desacopla o domínio das integrações externas (HubSpot API).
- **Validação e controle de fluxo:** implementados com verificações explícitas e exceções customizadas.

WebClient

- Utilizado para chamadas HTTP assíncronas à API da HubSpot.
- Preferido por ser não bloqueante e integrável com Spring WebFlux.

Redis para Armazenamento Temporário

- Utilizado para armazenar tokens de forma simples e eficiente.
- Alternativa leve para persistência temporária em comparação a um banco relacional.

RateLimiter (Resilience4j)

- Garante que não ultrapassemos os limites de requisição da API da HubSpot.
 - Fornece fallback resiliente, protegendo o sistema contra sobrecarga.
-

Bibliotecas Utilizadas

Biblioteca	Motivo
Spring Boot	Framework principal para construção da API
Resilience4j	Controle de falhas e limitação de taxa
WebClient (Spring)	Comunicação não bloqueante com APIs externas
Redis (via Spring Data)	Armazenamento rápido e temporário de tokens e estados OAuth
Lombok	Redução de boilerplate (getters/setters/constructors)
JUnit 5 + Mockito	Testes unitários com mocks e injeção de dependência
Docker/Docker Compose	Containerização e execução facilitada de dependências

Melhorias Futuras

Segurança

- Implementar autenticação via JWT para proteger os endpoints `/contacts`.
- Adicionar verificação do domínio do e-mail antes de criar um contato.

Monitoramento

- Incluir métricas e logs estruturados com ferramentas como Prometheus e Grafana.
- Logar taxa de uso da API HubSpot.

Refresh de Token Automatizado

- Automatizar o refresh de token quando este expirar, utilizando o `refresh_token` salvo no Redis.

Testes

- Adicionar testes de integração utilizando Testcontainers (Redis e WebClient mockado).
- Cobertura de testes com relatórios via Jacoco.

Painel Admin

- Criar um dashboard para visualizar contatos criados, tokens ativos e estatísticas de uso.

Integração

- Usar o payload recebido pelo Webhook para notificar com kafka ou afins.
- Persistir Contatos no banco relacional e manter sincronizado ao hubspot