



# Desafio Back end

## Desafio Técnico – Back-end na Lacrei Saúde

 Olá, futura pessoa voluntária de Back-end da Lacrei Saúde!

Estamos felizes por contar com você em nosso time voluntário! 

Sua contribuição será essencial para construir soluções que ampliam o acesso à saúde inclusiva e de qualidade para a comunidade LGBTQIAPN+.



### Sobre a Atividade Voluntária

- **Duração:** 3 meses
- **Carga horária:** 20 horas semanais
- **Encontros obrigatórios:**
  - 🕒 Segundas e terças, das 19h às 20h30 (horário de Brasília)



### Proposta do Desafio

## ➡ Desenvolva uma API funcional, segura e pronta para produção com propósito e impacto social.

Sua missão será desenvolver uma **API RESTful de Gerenciamento de Consultas Médicas**, com foco em:

- Qualidade de código
- Segurança dos dados
- Boas práticas de desenvolvimento
- Preparação para ambiente de produção

Este projeto será base para integrações com outros serviços da Lacrei Saúde, incluindo sistema de pagamentos, deploy e monitoramento.









## Sobre os dados e regras de negócio

- O cadastro de profissionais deve conter: Nome social, Profissão, Endereço, Contato.
- As consultas devem conter: Data, Profissional vinculado (chave estrangeira).
- Todos os retornos da API devem estar em **JSON**.




## O que esperamos de você — Itens obrigatórios:

- **CRUD completo:**
  - ◆ Profissionais da saúde
  - ◆ Consultas (relacionadas a profissionais)
  - ◆ Busca de consultas pelo ID do profissional
- **Segurança obrigatória:**
  - ◆ Sanitização e validação dos dados
  - ◆ Proteção contra SQL Injection
  - ◆ Implementação de CORS configurado corretamente
  - ◆ Controle básico de autenticação (ex.: API Key, JWT ou Token simples)
  - ◆ Logs de acesso e erros

-  **Tecnologias obrigatórias:**
    - Python com Django + Django REST Framework
    - Poetry (gerenciamento de dependências)
    - PostgreSQL
    - Docker (containerização)
    - GitHub Actions (para CI/CD)
  -  **Testes automatizados:**
    - ◆ Usando `APITestCase` do Django
    - ◆ Cobertura mínima:
      - CRUD de consultas
      - CRUD de profissionais
      - Testes de erro (ex.: requisição inválida, dados ausentes)
  -  **Deploy funcional:**
    - ◆ Ambientes separados: staging e produção (AWS)
  -  **Pipeline CI/CD:**
    - ◆ GitHub Actions com steps obrigatórios:
      - Lint
      - Testes
      - Build
      - Deploy
  -  **Documentação obrigatória:**
    - README com setup local e via Docker
    - Execução dos testes
    - Fluxo de deploy (CI/CD)
    - Justificativas técnicas das escolhas feitas
    - **Proposta de rollback funcional:**
      - Ex.: Deploy Blue/Green, Revert no GitHub Actions, Preview Deploy
  -  **(Bônus recomendado):**
    - Proposta de integração com a Assas (mock, arquitetura ou real)
    - Geração de documentação da API (ex.: Swagger, Redoc, Postman)
-



## Como irá fazer - Fluxo de atuação

1. Criar uma API Restful com:
    - Cadastro, edição, exclusão e listagem de profissionais da saúde
    - Cadastro e edição de consultas médicas com vínculo ao profissional
    - Busca por consultas utilizando o ID da pessoa profissional
  2. Garantir segurança e validação de dados:
    - Sanitização de inputs
    - Prevenção contra SQL Injection e outras vulnerabilidades
  3. Criar o projeto usando:
    - **Python**, com **Django + Django REST Framework**
    - **Poetry** para gerenciar dependências
    - **PostgreSQL** como banco de dados
    - **Docker** para configurar o ambiente
-  Configurando e Utilizando o Ambiente Docker para sua Aplicação
- **GitHub Actions** para automatização de testes e deploy
  4. Realizar o deploy da aplicação em:
    - **Staging e produção**, utilizando a AWS
  5. (Opcional mas será esta integração em que irá atuar) Propor uma **integração com a AssAs** para split de pagamento (pode ser mock ou proposta de fluxo com base na documentação pública).
  6. (Opcional, mas super valorizado!) Documentar um **fluxo de rollback** para o deploy da aplicação em caso de falha.



## Documentação e Testes

- Criar um **README completo**, contendo:
  - Setup do ambiente local e com Docker
  - Instruções para rodar o projeto
  - Instruções para rodar os testes com `APITestCase`
  - Explicações sobre decisões técnicas
  - Instruções de como foi feito o deploy (ambientes, ferramentas, fluxo CI/CD)

- Documentar no repositório os **erros encontrados, decisões e melhorias propostas** ao longo do desafio.



## Prazo de entrega

Você terá **5 dias ÚTEIS** após o recebimento deste desafio para finalizá-lo.

Envie o link do repositório público no GitHub (com README, deploy e documentação) para:



[desenvolvimento.humano@lacreisaude.com.br](mailto:desenvolvimento.humano@lacreisaude.com.br)



## CrITÉrios de Aceite

Item	Obrigatório	Observações
CRUD funcional de profissionais e consultas	✓	Incluindo busca por ID do profissional
Segurança (sanitização, CORS, autenticação)	✓	Proteção contra SQL Injection, API segura
Docker + PostgreSQL configurados	✓	Setup replicável para qualquer ambiente
GitHub Actions (CI/CD)	✓	Build, testes e deploy automatizados
Deploy funcional (staging e produção)	✓	Na AWS ou serviço equivalente
Testes unitários e de erro com APITestCase	✓	Cobertura mínima exigida
README completo + rollback	✓	Setup local, CI/CD, rollback e justificativas técnicas
Documentação da API (Swagger, Postman, etc.)	■ Opcional	Fortemente recomendado para profissionalização da entrega
Proposta de integração com Assas	■ Opcional	Agrega valor ao entendimento de fluxo de pagamentos



## O que você ganha com a atividade voluntária

- Participação em um projeto real de impacto social
- Networking com profissionais de diversas áreas
- Desenvolvimento técnico prático com tecnologias atuais
- Certificado de participação na Lacrei Saúde
- A experiência de construir tecnologia com propósito, inclusão e impacto real

## Nosso agradecimento

Na Lacrei Saúde, acreditamos que **código é cuidado**, e tecnologia pode transformar realidades.

Ficamos muito felizes com sua dedicação e vontade de contribuir com algo tão significativo.

Seu trabalho será parte da construção de um sistema que acolhe, respeita e protege.

---

| 🥰 Boa sorte! Estamos aqui torcendo por você 🚀🌈