

# Desafio Técnico

Vaga: Desenvolvedor Pleno | MeuAT

**Prazo: 4 dias corridos**

## Contexto

O MeuAT é um CRM agrícola que trabalha com dados geoespaciais de fazendas. Uma funcionalidade core do produto é a busca de fazendas por localização — seja por ponto exato ou por proximidade.

Seu desafio é construir uma API que permita consultar fazendas do estado de São Paulo usando queries geoespaciais.

## O Desafio

Construir uma API REST usando Python/FastAPI que permita buscar fazendas por localização geográfica, utilizando PostgreSQL com a extensão PostGIS.

## Requisitos Obrigatórios

### Stack:

- Python 3.10+
- FastAPI
- PostgreSQL + PostGIS
- Docker + Docker Compose

### Endpoints da API:

1. **GET /fazendas/{id}**  
Retorna os dados de uma fazenda específica pelo ID (CAR).
2. **POST /fazendas/busca-ponto**  
Recebe coordenadas (latitude/longitude) e retorna a(s) fazenda(s) que contém aquele ponto.  
Body: { "latitude": -23.5505, "longitude": -46.6333 }
3. **POST /fazendas/busca-raio**  
Recebe coordenadas + raio em quilômetros e retorna todas as fazendas dentro desse raio.  
Body: { "latitude": -23.5505, "longitude": -46.6333, "raio\_km": 50 }

### Infraestrutura:

1. Docker Compose que sobe todo o ambiente com um único comando
2. PostgreSQL com extensão PostGIS habilitada
3. Script de seed que carrega os GeoJSONs automaticamente ao iniciar o container
4. README com instruções claras de como rodar o projeto

**Expectativa:** Um `docker-compose up` deve ser suficiente para subir a aplicação funcionando, com dados já carregados.

### Dados de São Paulo:

<https://drive.google.com/file/d/15ghpnwzdDhFqelouqvQwXlbzovtPhlFe/view?usp=sharing>

## Bônus (diferenciais)

Não são obrigatórios, mas demonstram senioridade e atenção a qualidade:

1. **Testes automatizados** — pytest cobrindo os endpoints principais
2. **Documentação Swagger customizada** — além do auto-generated, com exemplos e descrições
3. **Paginação** — nos endpoints de busca
4. **Health check endpoint** — GET /health que verifica conexão com o banco
5. **CI básico** — GitHub Actions rodando lint + testes
6. **Filtros adicionais** — busca por nome da fazenda, filtro por área mínima/máxima
7. **Índices espaciais** — uso correto de índices GiST para otimizar queries
8. **Logs estruturados** — logging adequado das requisições

## Dados

Junto com este documento, você receberá um arquivo GeoJSON contendo polígonos de fazendas do estado de São Paulo. Use esses dados para popular o banco.

O seed deve ser executado automaticamente quando o container subir pela primeira vez.

## Entrega

- Repositório Git (GitHub, GitLab ou similar)
- README.md com instruções de como rodar
- Código fonte completo
- **Prazo: 4 dias corridos** a partir do recebimento deste desafio

## Critérios de Avaliação

Critério	Peso
Funciona com docker-compose up	<b>Eliminatório</b>
Uso correto do PostGIS (ST_Contains, ST_DWithin)	Alto
Estrutura e organização do código	Alto
Qualidade do README e documentação	Médio
Tratamento de erros e edge cases	Médio
Bônus implementados	Diferencial

## Observações Importantes

- Valorizamos código limpo e bem organizado mais do que features extras
- Se não conseguir completar tudo, entregue o que fez com um README explicando o que falta
- Commits bem escritos são um diferencial
- Dúvidas? Entre em contato — faz parte do processo saber perguntar

**Boa sorte!**