

Objetivo

Construir um **aplicativo que centraliza e gerencia oportunidades de trabalho freelancer**, permitindo:

- Buscar e filtrar **vagas reais** de várias plataformas (Excel, Python, Data Entry etc.);
 - Acompanhar **projetos aplicados / em andamento / concluídos**;
 - (Posteriormente) gerar **insights** — ganhos por mês, taxa de sucesso, categorias mais lucrativas etc.
-

1. Plataformas de freelancing com API oficial

Essas são as **melhores opções para integração direta**, porque oferecem endpoints documentados e estáveis:

Plataforma	API Oficial	Nível de Acesso	O que você pode obter	Observações
Upwork	 developers.upwork.com	OAuth2	Jobs, freelancers, contratos, mensagens, categorias	API completa, permite buscas e análises. Excelente ponto de partida.
Freelancer.com	 developers.freelancer.com	API Key + OAuth	Listar projetos, detalhes, bids, mensagens, perfis	Extremamente completa. Boa para montar um agregador.
Fiverr	 Limitada (apenas "Fiverr Business")	Apenas via parcerias	Informações limitadas sobre gigs e sellers	Sem API pública, só via acordo empresarial.
Toptal	 Não possui	Nenhum	—	Plataforma fechada, sem API e scraping difícil (proteções JS).

Plataforma	API Oficial	Nível de Acesso	O que você pode obter	Observações
PeoplePerHour	🚫 Antiga API descontinuada	Nenhum	—	Scraping leve possível, mas requer login.
Guru	🚫 Não pública	—	—	Pode ser raspada com cautela, estrutura HTML estável.
Workana	🚫 Não pública	—	—	Fácil de raspagem, HTML limpo, ótimo para projetos em PT/ES.
99Freelas	🚫 Não pública	—	—	Plataforma brasileira, fácil de scraping. Ótima fonte para jobs em português.

🕸️ 2. Plataformas com scraping viável (teoricamente simples)

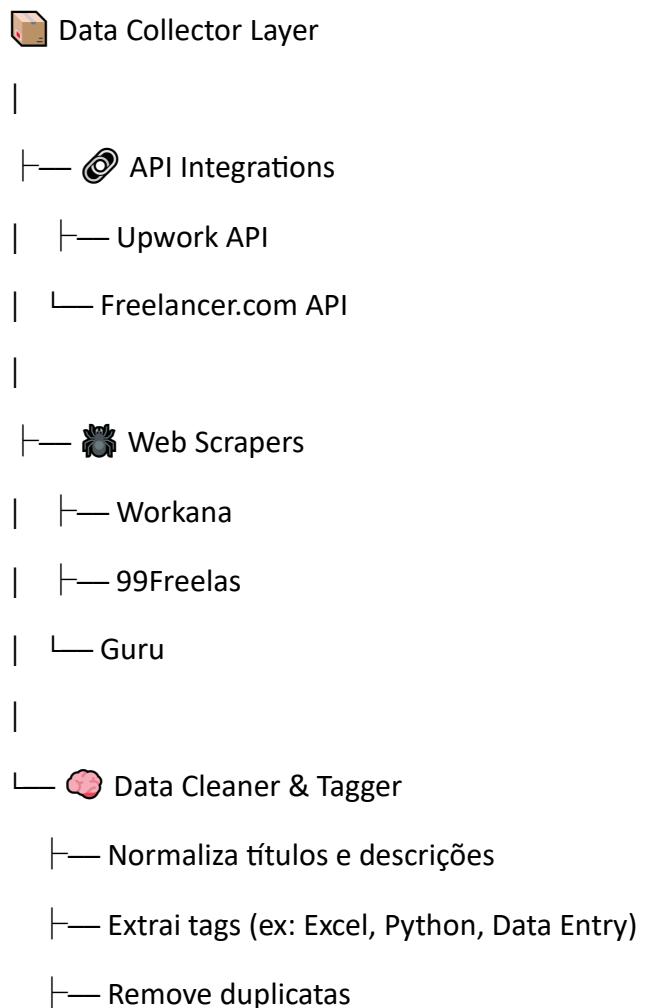
Essas são plataformas que **não oferecem API oficial**, mas cuja estrutura HTML é simples o suficiente para você construir um coletor (scraper) estável, **desde que respeite os limites éticos e legais** (ex: não fazer scraping autenticado, respeitar robots.txt e intervalos).

Plataforma	Tecnologias que funcionam bem	Dificuldade Técnica	Observações
Workana	Requests + BeautifulSoup	🟢 Fácil	HTML estável, campos de título, descrição e orçamento bem definidos.
99Freelas	Playwright (JS dinâmico)	🟡 Média	Requer renderização JS, mas estrutura previsível.

Plataforma	Tecnologias que funcionam bem	Dificuldade Técnica	Observações
Guru	Requests + BeautifulSoup	● Fácil	Permite capturar título, preço e tags.
PeoplePerHour	Playwright / Selenium	● Média	Algumas seções exigem login; requer mais cuidado.

3. Abordagem híbrida (API + scraping)

Você pode montar uma arquitetura combinando o melhor dos dois mundos:



Depois você grava tudo em um banco como **PostgreSQL** e exibe via painel (React, Streamlit, etc.).

4. Dados que você pode coletar (para filtros e gestão)

Aqui estão os campos mais úteis para armazenar e filtrar:

Campo	Fonte (API/Scraping)	Uso
platform	Todas	Saber de qual site veio o job
title	Todas	Filtro principal
description	Todas	Análise de palavras-chave
budget_min / budget_max	Upwork, Freelancer, Workana	Filtrar por valor
category / skills	Upwork, Freelancer	Criar tags (ex: Excel, Data Entry)
posted_date	Todas	Mostrar vagas recentes
url	Todas	Acesso direto
status (aplicado, em andamento, concluído)	Manual ou integração	Gestão pessoal

💡 5. Stack técnica sugerida

Camada	Tecnologia
Backend (API)	FastAPI (Python)
Scrapers	Scrapy ou Playwright
Banco de dados	PostgreSQL (ou Supabase, se quiser hospedagem fácil)
Frontend	React (Next.js) ou Streamlit
Agendamento	Celery + Redis (para coletar dados periodicamente)
Hospedagem	Railway / Render / Vercel (grátis no início)

💡 6. Ideias futuras para tornar o app único

- ⏸ **Alertas automáticos:** notificação quando aparecer um job com tags específicas (ex: “Excel”, “Python”, “VBA”).

- **Dashboard pessoal**: evolução de ganhos, taxa de sucesso, tempo médio de resposta.
- **Classificador inteligente**: usar IA leve (por exemplo, *scikit-learn* ou *spaCy*) para identificar automaticamente a categoria de cada job.
- **Histórico de aplicações**: registrar em qual job você aplicou, data, resultado, feedback do cliente.
- **Extensão Chrome (futuro)**: botão para salvar uma vaga diretamente no app.