

# TDAI - Visão Geral e Direção Futura

## Visão Principal:

"Uma plataforma que entende produtos, melhora com o tempo, ajuda humanos a vender melhor e automatiza o que ninguém quer fazer."

---

## O Que Já Temos:

- Backend com FastAPI, estruturado.
- Upload de arquivos CSV, Excel e PDF funcionando.
- Parsing e salvamento de produtos com dados brutos.
- Pipeline de enriquecimento com scraping (Playwright), extração de texto (Trafilatura), e geração com OpenAI.
- Painel frontend simples (HTML all-in-one).
- Contagem de IA, planos, e controle de uso.

---

## Objetivo de Curto Prazo (MVP):

- Finalizar o frontend funcional.
- Permitir upload, listagem, enriquecimento e visualização dos produtos.
- Mostrar score e log por produto.
- Entregar uma experiência simples, direta e funcional.

---

## Estratégia de Evolução Progressiva para Agentes:

- Não interromper o que já está funcionando.
- Adaptar as funções atuais com formato de agente: (input\_data, context) -> output
- Criar uma orquestração fake (sem CrewAI) para simular pipeline de agentes.
- Cada etapa da pipeline registra log e altera o contexto.
- Preparar essas funções para futura integração com CrewAI.

---

Agentes Planejados:

- SearchAgent: busca URLs relevantes com base no nome do produto.
- ScraperAgent: coleta HTML bruto das URLs.
- MetadataExtractorAgent: extrai dados estruturados (JSON-LD, OpenGraph).
- TextExtractorAgent: extrai corpo de texto principal da página.
- LLMEnricherAgent: gera descrição, títulos, atributos com LLM.
- ValidatorAgent: avalia qualidade, coerência e completude dos dados.
- CommanderAgent: coordena fluxo, ativa reproprocessamento ou fallback.
- RecoveryAgent: tenta URLs alternativas, busca em cache ou gera fallback.

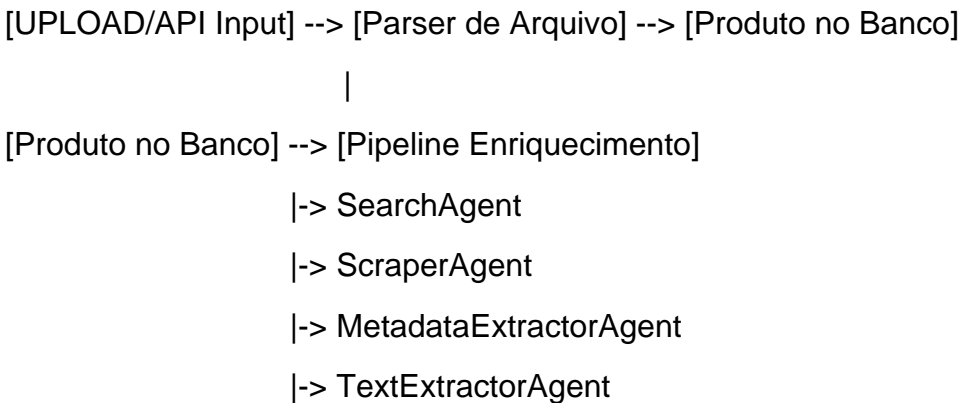
---

Melhorias Futuras:

- Score automático de qualidade por produto.
- Feedback do usuário integrando com IA.
- Aprendizado contínuo baseado em uso real e resultados.
- Multi-versões de títulos/descrições para diferentes canais.
- Exportação e integração com lojas, marketplaces e ERPs.
- Dashboards com insights sobre a performance do catálogo.

---

Fluxograma - Backend:



|-> LLMEnricherAgent  
|-> ValidatorAgent  
|-> Commander/RecoveryAgent  
--> [Produto Atualizado]

---

Fluxograma - Frontend:

[Dashboard Inicial] --> [Upload de Arquivo] --> [Tabela de Produtos]  
--> [Ver Detalhes Produto]  
--> [Ver Score, Títulos, Descrição, Logs]  
--> [Editar/Refazer com IA]  
--> [Plano e Créditos]

---

Fluxograma - Aplicação Geral:

[Usuário Frontend]	[Sistema Backend]
Upload/Visualização	-----> Recebimento & Parsing
Comando de Ação (Enriquecer)	-----> Pipeline Inteligente (Agentes)
Visualiza Resultado	<----- Produto Atualizado + Logs

---

Visão de Longo Prazo:

- TDAI se torna o "motor semântico" de qualquer catálogo.
- IA aprende e melhora com os dados de todos os clientes.
- Produto vira um ecossistema de geração, otimização e publicação de conteúdo.
- Solução plugável com painel, API e assistente inteligente.

---

Próxima Etapa:

- Finalizar frontend completo.
- Ligar o painel aos dados reais do backend.
- Implementar pipeline simulada de agentes com log estruturado.

Responsável: Julio

Acompanhamento: ChatGPT (CrewAI futuro)