

# Relatório Final de Projeto: Tribut.AI

Nome do Grupo: InovaIA

Integrantes do Grupo:

- Abraão Moreira
- Verônica Barros

## 2. Descrição do Tema Escolhido

O **TRIBUT.AI** é um agente virtual baseado em Inteligência Artificial, especializado na **Reforma Tributária (Emenda Constitucional n.º 132/2023)**. O objetivo é atuar como um consultor de apoio, fornecendo **respostas rápidas, seguras e contextualizadas**, fundamentadas por uma base de conhecimento atualizada.

O projeto **TRIBUT.AI** foi estruturado para resolver a **complexidade e a incerteza** inerentes ao período de transição da reforma, focando em:

1. **Busca Contextual (RAG):** Responder a dúvidas complexas com citação obrigatória da fonte legal.
2. **Análise Qualitativa:** Realizar simulações de impacto e análise estratégica com base em códigos fiscais (NCM/CFOP), sem projetar valores numéricos indefinidos.
3. **Rastreabilidade:** Manter logs completos de uso para medir o engajamento e identificar temas recorrentes.

## 3. Público Alvo

A solução é voltada para profissionais que precisam de validação rápida e análise estratégica de impacto fiscal:

- **Auditores e Analistas Fiscais:** Para validar a aplicação das novas regras e buscar embasamento legal.
- **Pequenos Escritórios Contábeis:** Para suporte na interpretação da legislação e na transição de regimes.
- **Gerentes Comerciais e de Produto:** Para análise estratégica de impacto fiscal em produtos e operações.

## 4. Justificativa do Tema Escolhido e Valor Agregado

A reforma tributária é o maior desafio fiscal da próxima década. A justificativa do **TRIBUT.AI** se baseia em:

- **Necessidade de Mercado:** Pesquisas indicam baixa adoção de IA em processos fiscais complexos e alta demanda por automação na interpretação legal (48% dos entrevistados gastam tempo significativo com tarefas manuais).
- **Valor Agregado (Confiança e Velocidade):**
  - **Redução de Erros:** Fornece respostas com citação da base legal, diminuindo a dependência de interpretações manuais.
  - **Agilidade na Decisão:** Reduz o tempo gasto na interpretação e aplicação das novas regras tributárias.
  - **Análise Estratégica:** Transforma códigos fiscais (NCM/CFOP) em **análise de impacto qualitativo**, alertando sobre regimes diferenciados e regras de crédito/débito.

## 5. Detalhamento do que Foi Desenvolvido

O projeto foi implementado com uma arquitetura de Microserviço Híbrido, dividindo as responsabilidades entre um *backend* robusto (Python) e um *frontend* ágil (Next.js).

### 5.1. Arquitetura e Tecnologia

- **Backend (Lógica do Agente): FastAPI (Python).** Hospeda a lógica do Agente, o RAG e os *endpoints* da API. Utiliza a biblioteca **LangChain** como *framework* de orquestração para todas as operações de Agente, RAG e Memória. A estrutura de rotas é a seguinte:
  - **Módulo de Chat (/api/v1/chat):** Gerencia a interação em tempo real com o Agente (RAG) e a recuperação de informações. **O funcionamento dos chats é baseado em services que operam a Chain RAG e o LLM, utilizando o modelo Gemini (Google) para a geração de respostas e análise de contexto.**
  - **Módulo de Sessões (/api/v1/sessions):** Gerencia a criação e o estado das conversas (*session\_id*).
  - **Módulo de Base de Conhecimento (/api/v1/documentos):** Gerencia a ingestão, categorização e manutenção dos documentos.
    - **Ingestão de Legislação (PDF/TXT):** Processamento de PDFs e textos com *chunking* otimizado para Artigos e Metadados de Autoridade/Vigência.
    - **Ingestão de Notas Fiscais (XML):** O sistema realiza um **pré-processamento** para extrair dados importantes (NCM, CFOP, Valor Total) para serem usados como **Metadados** (filtros de busca) antes de indexar o resumo do documento no Vector Store.
  - **Módulo de Dashboard/Métricas (/api/v1/dashboard):** Fornece dados agregados para o dashboard e relatórios de uso.

- **Frontend (Interface): Next.js (TypeScript)**, utilizando **Tailwind CSS** com a biblioteca de componentes **DaisyUI** para criar uma interface gráfica do *backend*, interativa e responsível.
- **Vector Store (Base de Conhecimento): ChromaDB**. Armazena os *embeddings* de todos os documentos ingeridos em uma única **Collection**. O *embedding* é realizado utilizando modelos de *language model* do Hugging Face (bibliotecas Python). A separação lógica e a otimização da busca são feitas **exclusivamente através dos Metadados** (*Autoridade*, *Data\_Vigencia*, etc.).
- **Banco de Dados (Persistência): SQLite**. Usado para a gestão de **Sessões de Chat** e **Logs de Uso**, além de armazenar metadados detalhados de todos os documentos ingeridos (caminho do arquivo, NCMs/CFOPs extraídos e o conteúdo de referência que é indexado no Vector Store).

5.2. Funcionalidades-Chave Implementadas (MVP)

O Agente ACFI opera com as seguintes funções, garantindo o cumprimento dos Prompts:

Funcionalidade	Detalhamento	Requisito Atendido
Busca Contextual (RAG)	Sistema RAG com indexação de documentos (PDFs, TXT) e <i>chunking</i> otimizado. O Agente utiliza a informação para fundamentar a resposta.	Chat Inteligente, Busca Contextual.
Citação de Fonte	No <i>backend</i> , o LLM é instruído a usar <b>apenas</b> o contexto recuperado e citar o <b>Metadado</b> do documento (Ex: EC 132/2023, Artigo X).	Citação Obrigatória da Fonte (RF-003).
Memória Conversacional	Integração com o SQLite para persistir o histórico de mensagens por <i>session_id</i> . O Agente mantém o contexto da conversa.	Coerência da Conversa.
Roteamento de Busca	Utilização de <b>Metadados ricos</b> ( <i>Autoridade</i> , <i>Data_Vigencia</i> ) para filtrar a única Collection do ChromaDB e direcionar a busca com precisão.	Otimização do RAG.

<b>Captura de Dados Fiscais (NCM/CFOP)</b>	O sistema aceita o input do <b>XML</b> e faz o <b>parsing</b> para extrair os códigos fiscais <b>NCM</b> e <b>CFOP</b> . O Agente utiliza esses códigos como chave de busca para gerar uma <b>análise de regras e impacto qualitativo</b> da Reforma Tributária.	<b>Análise Estratégica (Regras).</b>
<b>Logger de Uso (Captura)</b>	Implementação de um Logger que registra no SQLite a <b>query</b> , a <b>final_answer</b> e a <b>fonte citada</b> de cada interação.	<b>Métricas Básicas de Sucesso</b> (Dados para Relatórios).
<b>Geração de Relatórios e Métricas</b>	Criação de interface para visualizar métricas de uso (consultas por tema, volume) e <b>dados agregados de Notas Fiscais</b> (ex: volume de CFOPs analisados e valor total transacionado se disponível no log).	<b>Dashboard de Uso</b> , Métricas de Sucesso.

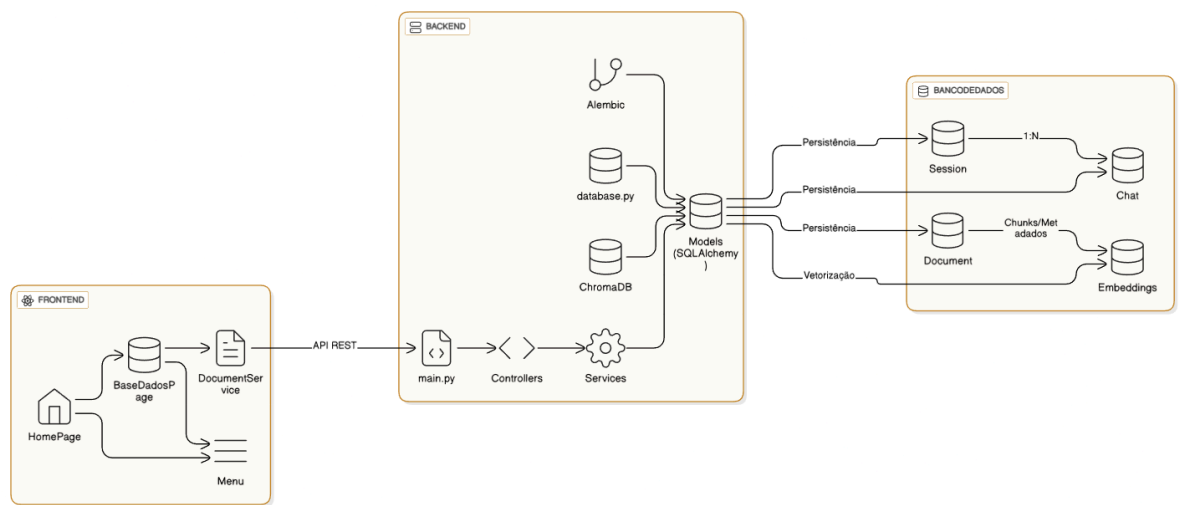
### 5.3. Comportamento do Agente

O comportamento do Agente é definido por instruções diretas na Chain RAG, garantindo o tom consultivo e o foco nas análises:

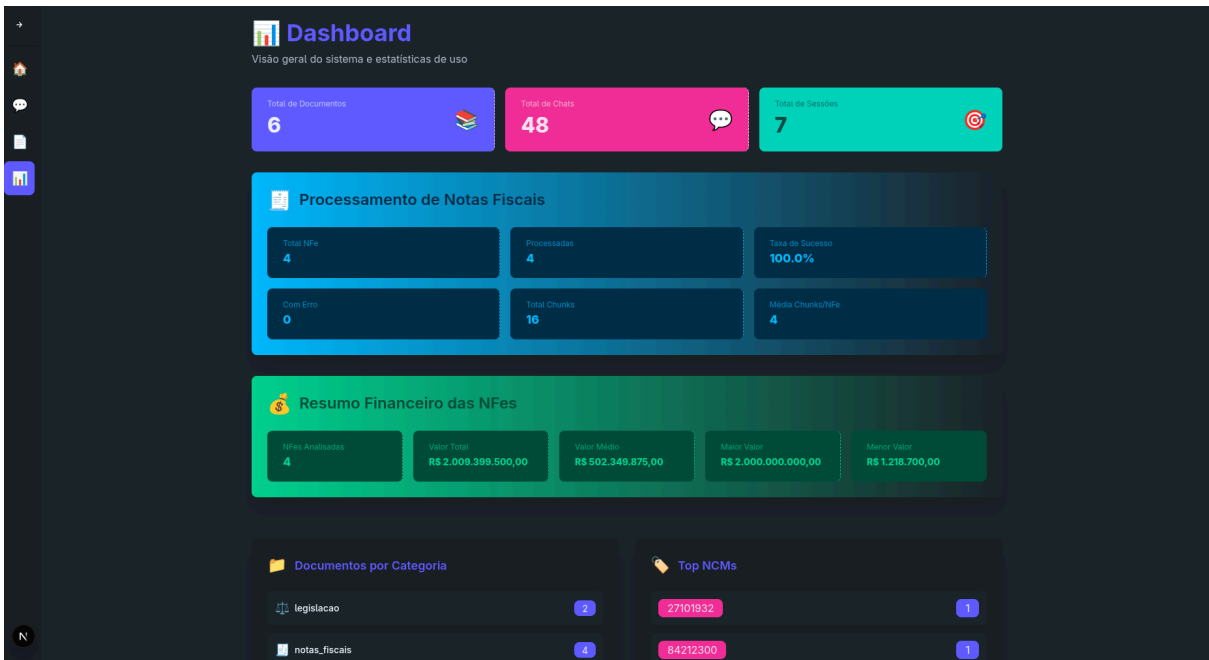
- **Tom Consultivo:** Responde dúvidas com resumo legal, citação obrigatória e recomendação de conformidade.
- **Análise Estratégica:** Foca em **análise de impacto (qualitativa)** com base em NCM/CFOP, sem calcular valores incertos.
- **Interpretação Simples:** Explica artigos legais em linguagem simples, reforçando o alerta de verificação em fontes oficiais.

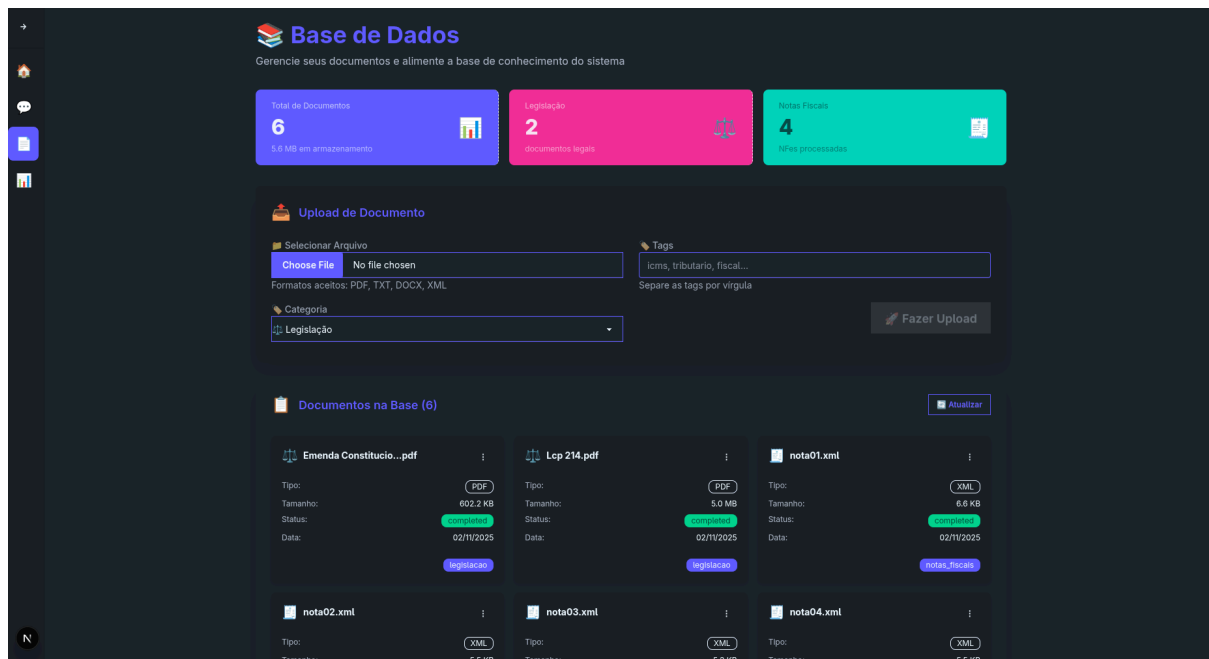
# 6. Elementos Adicionais: Diagrama de Arquitetura

## Diagrama Simplificado de Arquitetura (MVP)



eraser





## 7. Link para o Repositório do GitHub

Link: <https://github.com/AbraaoM/ACFI>