



Objetivo do Projeto

Lexi um chatbot baseado em inteligência artificial com foco no Direito Brasileiro, desenvolvido para auxiliar nos estudos para concursos públicos. O objetivo principal permitir que estudantes faam perguntas em linguagem natural e recebam respostas precisas, baseadas em uma base de dados composta por documentos legais e materiais de estudo (em PDF).

Estrutura do Projeto

```
project/
├── app/
│   ├── main.py           # API FastAPI com rotas para upload e perguntas
│   ├── qa_pipeline.py    # Pipeline de perguntas com LangChain + OpenRouter
│   ├── document_loader.py # Carregamento e indexação dos PDFs
│   ├── config.py         # Configurações (chaves, diretórios, etc.)
│   └── vector_store/     # Banco vetorial local (ChromaDB)
├── data/
│   └── pdfs/             # PDFs enviados pelo usuário
├── requirements.txt      # Bibliotecas necessárias
└── README.md            # Documentação do projeto
```

Tecnologias Utilizadas

- LangChain - Framework para construir pipelines com LLMs.
- ChromaDB - Banco vetorial local para armazenar embeddings dos documentos.
- OpenRouter - Intermediário de modelos LLM, usado para acessar o modelo Mixtral-8x7B.
- FastAPI - Framework web para criar a API.
- PyMuPDF - Biblioteca para leitura de arquivos PDF.
- Python-dotenv - Gerenciamento de variáveis de ambiente.
- Uvicorn - Servidor ASGI para rodar a API.

Funcionamento Esperado

1. Envio de Documentos:

- O usuário envia arquivos PDF para a API via endpoint /upload.
- Esses arquivos processados, extraídos e convertidos em embeddings vetoriais com LangChain + OpenAIEmbeddings.
- Os vetores serão armazenados localmente no banco vetorial ChromaDB.

2. Realização de Perguntas:

- O usuário envia uma pergunta via endpoint.
- O sistema busca nos documentos os trechos mais relevantes usando recuperação semântica

(similaridade vetorial).

- A pergunta e os documentos relevantes serão enviados ao modelo Mixtral-8x7B via OpenRouter. - O modelo responde com base no conteúdo recuperado, retornando uma resposta precisa.

Casos de Uso

- Estudantes de Direito que desejam revisar leis e jurisprudência de forma conversacional.
- Candidatos a concursos públicos que queiram reforçar o estudo por temas. - Leitura assistida de PDFs com possibilidade de perguntas sobre o conteúdo.

Próximos Passos

- Criar interface web amigável (frontend com cores da identidade visual rosa).
- Adicionar autenticação de usuários.
- Suporte para múltiplos arquivos e histórico de perguntas.
- Otimização de desempenho com banco vetorial hospedado ou em memória. - Deploy em servidor (VPS ou serviço em nuvem como Render, Railway etc.).

DEMANDAS

- Conectar modelo LLM (Mixtral-8x7B) via OpenRouter
- Estrutura básica do projeto com FastAPI, LangChain e Chroma
- Organização dos documentos em /data/pdfs
- Configuração de variáveis com .env
- Criar frontend web responsivo com identidade visual rosa
- Implementar autenticação de usuários
- Adicionar histórico de perguntas/respostas por usuário
- Adicionar suporte para múltiplos arquivos por sessão
- Criar painel de administração (opcional)
- Implementar sistema de logs e métricas de uso
- Deploy do sistema (Render, Railway, VPS etc.)
- Criar landing page com explicação do projeto
- Adicionar feedback de avaliação nas respostas
- Estrutura inicial do projeto criada
- Rotas de upload e pergunta implementadas
- Leitura e indexação automática de PDFs com LangChain
- Integração com embeddings OpenAI e banco vetorial Chroma
- Teste de funcionalidade com API funcionando localmente
- Documento explicativo e versão em PDF criada
- Identidade visual inicial com logo e interface mockup