

## Documentação Detalhada

### Visão Geral

Este documento detalha o funcionamento do chatbot inteligente baseado em conteúdo de PDFs, implementado em Python.

### Objetivo

Permitir ao usuário interagir com documentos PDF através de perguntas, recebendo respostas contextuais baseadas no conteúdo dos documentos.

### Arquitetura do Projeto

#### 1. Extração de Texto

- Utiliza PyMuPDF (fitz) para ler e extrair o texto dos PDFs.

#### 2. Processamento de Texto

- O texto extraído é fragmentado em pedaços de tamanho controlado (500 caracteres) com sobreposição.

#### 3. Geração de Embeddings

- Cada fragmento é convertido em um vetor numérico usando OpenAIEmbeddings.

#### 4. Banco de Vetores

- Os vetores são indexados em uma base FAISS para busca vetorial de alta performance.

#### 5. Módulo de Perguntas e Respostas

- O RetrievalQA da LangChain integra o modelo LLM com o indexador vetorial.

#### 6. Interface de Usuário

- Modo CLI: interação via terminal com app.py.
- Modo Web: interface Streamlit em app\_streamlit.py.

### Detalhamento de Arquivos

- app.py: Versão de linha de comando.
- app\_streamlit.py: Versão de interface web.
- inputs/texto.txt: Arquivo de entrada para testes.
- docs/DOCUMENTATION.md: Este documento.

### Configuração de Ambiente

1. Configure a variável de ambiente: `export OPENAI_API_KEY=<sua_chave>`.
2. Instale as dependências com: `pip install streamlit langchain openai pymupdf faiss-cpu`.

### Executando o Projeto

- CLI: `python app.py`
- Web: `streamlit run app_streamlit.py`

### Possíveis Expansões

- Adicionar autenticação de usuário.
- Suporte a múltiplos formatos de documento.
- Deploy em nuvem (Azure/AWS/GCP).
- Monitoramento de uso e métricas de interação.

### Contato

- GitHub: <https://github.com/rafaelqueiroz>
- LinkedIn: <https://www.linkedin.com/in/rafaelqueiroz>