

# Documentação do Projeto – Chatbot Nascentia

**Projeto:** Chatbot Nascentia

**Curso:** Ciência de Dados & Machine Learning – CEUB

**Integrantes:** Rafael Martins, Felipe Yoshida, Mateus Bitar, João Pedro Borges, José Muller e Mateus Alves.

**Ambiente:** VS Code + Python 3.10.8

---

## Sumário

1. Apresentação do Problema
  2. Descrição dos Dados
  3. Metodologia e Processo Analítico
  4. Resultados e Evidências Visuais
  5. Estrutura do Projeto e Scripts
  6. Considerações Finais, Aprendizados e Recomendações
  7. Link para o Repositório no GitHub
- 

## 1. Apresentação do Problema

A empresa Nascentia atua com educação pré-natal e profissional, oferecendo cursos online para gestantes e enfermeiros. Também presta serviços de assistência pré-natal, parto, pós-parto, consultoria em amamentação e acupuntura para gestantes.

O problema identificado foi a dificuldade de responder dúvidas frequentes sobre os serviços e conteúdos oferecidos, de forma precisa e sem gerar respostas genéricas ou incorretas. Para isso, desenvolvemos um chatbot inteligente baseado em documentos técnicos da própria Nascentia, com garantia de rastreabilidade das fontes.

---

## 2. Descrição dos Dados

Todos os dados utilizados são documentos em PDF fornecidos pela empresa Nascentia. Eles abordam temas como:

- Procedimentos de enfermagem
- Cuidados no pré-natal e parto
- Plano de parto
- Serviços oferecidos
- Equipe da Nascentia
- Cursos e mentorias

Os PDFs foram colocados na pasta `data/`. Esta pasta possui uma amostra de 3 PDFs como exemplo.

Além disso, códigos auxiliares também tratam documentos `.docx` com o seguinte pipeline:

- Extração de texto com `python-docx`
- Limpeza de campos e caracteres
- Conversão final para PDF com a biblioteca `fpdf`

---

## 3. Metodologia e Processo Analítico

### Ambiente de Desenvolvimento:

- Visual Studio Code
- Python 3.10.8

**Ambiente Virtual:** Foi criado um ambiente virtual para isolar as dependências do projeto. Recomenda-se a criação de um ambiente virtual com os seguintes comandos:

```
python -m venv venv
```

```
source venv/bin/activate # Linux/Mac
```

```
venv\Scripts\activate # Windows
```

`pip install -r requirements.txt`

**Arquivo `.env`:** A chave da API da OpenAI deve ser colocada no arquivo `.env`, no seguinte formato:

`OPENAI_API_KEY=sua-chave-aqui`

A chave da OpenAI é paga, portanto é necessário possuir uma conta com créditos válidos.

#### **Principais Bibliotecas Utilizadas:**

- `langchain`, `langchain-core`, `langchain-community`, `langchain-openai`
- `faiss-cpu` para vetorização e recuperação semântica com persistência local
- `pypdf` e `PyPDFLoader` para leitura dos PDFs
- `dotenv`, `openai`, `ChatOpenAI`, `ConversationalRetrievalChain`

#### **Técnica Principal:** RAG (Retrieval-Augmented Generation)

- O usuário faz uma pergunta
- O sistema busca trechos relevantes nos PDFs usando FAISS + Embeddings OpenAI
- Um prompt estruturado força o modelo a responder apenas com base no contexto recuperado
- Cada afirmação precisa indicar a fonte [`Fonte: nome-do-pdf p.X`]

#### **Prompt Especializado:**

- Impede alucinações, suposições e informações externas
- Rejeita perguntas sem resposta nos documentos com "Não tenho informações sobre isso nos documentos analisados."

#### **Memória:**

- `ConversationBufferMemory` preserva o histórico da conversa para perguntas encadeadas.

---

## **4. Resultados e Evidências Visuais**

#### **Prints do Terminal:**

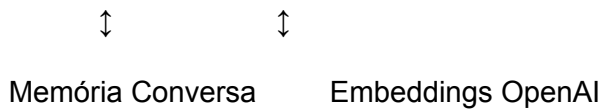
- O chatbot responde com alta precisão e segurança
- Toda resposta traz a fonte do documento original

#### **Testes de Robustez:**

- Roteiro com perguntas complexas e "pegadinhas" para testar alucinações
- Todas as alucinações foram evitadas graças ao controle rígido do prompt

## Arquitetura da Solução:

Usuário ↔ LangChain ChatPrompt ↔ Vetor FAISS ↔ PDF Nascentia



## Exemplo de conversa:

✓ Chatbot RAG com memória pronta! (Digite 'sair' para encerrar)

Você: o que é a Nascentia?

Assistente: A Nascentia é uma empresa especializada em educação pré-natal e profissional, oferecendo cursos online voltados tanto para gestantes quanto para enfermeiros. Além dos cursos, a empresa disponibiliza serviços de terapia de acupuntura para gestantes, assistência médica e de enfermagem durante o pré-natal, parto e pós-parto, bem como consultoria em amamentação. O nome "Nascentia" deriva do Latim, significando "Nascimento", e a empresa acredita que o nascimento de um bebê é um rito de passagem que transforma não apenas as pessoas, mas também suas relações com os outros e com o mundo. A Nascentia valoriza um nascimento respeitoso, seguro e gentil, que favorece a criação de vínculos fundamentais para a formação do ser. A assistência prestada pela equipe da Nascentia é fundamentada em evidências científicas e princípios humanísticos, buscando sempre um cuidado centrado na família, respeitando suas crenças e valores, e promovendo a incorporação do mundo do Sagrado na experiência do nascimento. [Fonte: O que é a Nascentia.pdf p.1]

Você: quais são os cursos online?

Veja que o Chatbot explica detalhadamente o que é a Nascentia e no fim especifica a fonte com o nome do arquivo que ele pegou essa informação.

## 5. Estrutura do Projeto e Scripts

### Pasta principal de código: `src/`

Todos os scripts do projeto estão contidos nesta pasta:

- `chatbot.py`: Script principal do chatbot inteligente RAG com PDFs e memória
- `chatbot_basico.py`: Versão de teste mais simples, sem memória nem prompt especializado
- `testar_api.py`: Script auxiliar para verificar se a API da OpenAI está funcionando corretamente
- `script_código_tratamento_docx.ipynb`: Notebook com código que trata arquivos `.docx` e os converte em PDF para futura inclusão na base de conhecimento

### Outras pastas relevantes:

- `data/`: onde ficam os arquivos PDF usados como base de conhecimento
- `faiss_index/`: diretório que armazena os vetores FAISS com persistência local

- `.env`: arquivo com a variável `OPENAI_API_KEY` (não deve ser versionado)
- `Docs/`: diretório com documentos sobre o projeto
- `requirements.txt` arquivo com todas as dependências necessárias para o projeto.

\* **Importante:** toda vez que for adicionar novos PDFs no diretório de `data/`, é necessário excluir o diretório do `faiss_index`, pois ele estará desatualizado. Após rodar o script do `chatbot.py`, ele criará um novo `faiss_index` atualizado e persistido.

#### Versões Importantes:

- Python: 3.10.8
- LangChain: 0.3.25
- FAISS: 1.11.0
- OpenAI: 1.78.0

---

## 6. Considerações Finais, Aprendizados e Recomendações

- Evitar alucinações exige controle rígido do prompt e do contexto.
- A integração do FAISS com LangChain foi essencial para boa performance.
- O projeto pode ser facilmente expandido com mais PDFs.
- Em ambiente real, a interface pode ser portada para web via `Streamlit` ou `FastAPI`.

#### Recomendações Futuras:

- Criar uma interface amigável e personalizada com o `Streamlit`.
- Refinar mais ainda o chatbot se necessário.

---

## 7. Link para o Repositório no GitHub

 <https://github.com/FelipeYoshidaCEUB/chatbot-ceub>