

PASSO-A-PASSO

Streamlit

Importando as bibliotecas:

```
import streamlit as st
from streamlit_option_menu import option_menu
from streamlit_chat import message
from rag_model import generate
```

Mudando o nome na aba do browser

```
st.set_page_config(
    page_title="GPLux",
    layout="wide",
)
```

Definindo a seção de perguntas

```
# Função para exibir a seção de perguntas
def perguntar():
    st.header(":orange[Faça uma pergunta!]", divider='orange')
    st.write("")
    st.write("")
    st.subheader("Eu sou um software capaz de responder perguntas baseadas em seus dados.")

# Verificar se o estado da sessão já existe, caso contrário, inicializá-lo
if 'messages' not in st.session_state:
    st.session_state.messages = []

# Caixa de texto para inserir a pergunta
question = st.text_input("Digite aqui:")

# Botão para enviar a pergunta
if st.button("Enviar"):
    if question:
        # Chamar a função do modelo RAG para obter a resposta
        response = generate(question)
        # Armazenar a pergunta e a resposta no estado da sessão
        st.session_state.messages.append({"user": question, "bot": response})
        # Limpar a caixa de texto para nova entrada
        st.experimental_rerun()
    else:
        st.warning("Por favor, digite uma pergunta.")

# Exibir todas as mensagens no estilo chat
for msg in st.session_state.messages:
    st.markdown(f'<div style="background-color:#E0E0E0; padding:10px; border-radius:10px; margin:10px 0; text-align:right; display:inline-block; max-width: 70%; float:right;"><strong>Pergunta:</strong><br>{msg["user"]}</div>',
                unsafe_allow_html=True)
    st.markdown(f'<div style="background-color:#FFAE6B; padding:10px; border-radius:10px; margin:10px 0; text-align:left; display:inline-block; max-width: 70%; float:left;"><strong>Resposta:</strong><br>{msg["bot"]}</div>',
                unsafe_allow_html=True)
```

Título da página inicial

```
# Título da aplicação
st.title("Lux Logistics – Análise Logística")
```

Menu lateral

```
# Menu
with st.sidebar:
    selected = option_menu(None, ["Página Inicial", "Perguntar", "Power BI", 'GitHub'],
        icons=['house-heart', 'chat-square-text', 'graph-up', 'github'], menu_icon="map",
        default_index=0,
        styles={
            "container": {"padding": "0!important", "background-color": "#fafafa"},
            "icon": {"color": "orange", "font-size": "25px"},
            "nav-link": {"font-size": "18px", "text-align": "left", "margin": "0px", "—
                hover-color": "#eee"},
            "nav-link-selected": {"background-color": "gray"},
        }
    )
selected
```

Lógica para acesso pelo menu

```
# Lógica para o menu
if selected == "Página Inicial":

elif selected == "Perguntar":

elif selected == "Power BI":

elif selected == "GitHub":
```

Acesso a página inicial e texto

```
if selected == "Página Inicial":
    st.header(":orange[Bem-vindo ao nosso sistema de análise logística!]")
    st.write("")
    st.write("")
    st.subheader("A API (Aprendizagem por Projetos Integrados)", divider='orange')
    st.write("")
    st.markdown("""
Nosso projeto de API do 3º semestre de logística noturno, intitulado: __Análise
de Produtividade de Veículos, Otimização de Distribuição e Custo
de Rotas__, visa melhorar a eficiência operacional e reduzir custos de uma
empresa embarcadora de carga através de uma análise abrangente da
produtividade de seus veículos, otimização de distribuição e gestão de custos de
rotas; para desenvolver este projeto utilizamos a metodologia
ágil promovendo a proatividade, autonomia, colaboração e entrega de resultados
dos estudantes envolvidos.

Nosso objetivo foi implementar soluções baseadas em dados que otimizem o
transporte de carga e minimizem as despesas logísticas, para isso criamos
um banco de dados em SQL com as bases modeladas e enriquecidas em Python, com
indicadores e KPIs pertinentes ao projeto, aplicamos o método dos
transportes para a otimização da distribuição e construímos um dashboard em
Power BI para visualizar graficamente nossos resultados e indicadores.
Esperamos assim alcançar um aumento na produtividade dos veículos, otimização da
distribuição de cargas, redução significativa nos custos das rotas e
```

melhorar a gestão e visualização de indicadores de desempenho através do Power BI.

Este projeto representa uma abordagem inovadora para a gestão de logística, proporcionando uma visão clara e acionável para a tomada de decisões estratégicas.

Inclusão de imagens na página inicial

```
st.write("")
st.subheader("A Equipe", divider='orange')
st.write("")
with st.container():
    col1, col2, col3, col4 = st.columns(4)
    with col1:
        st.markdown(":orange[**Product Owner**]")
        st.image('rebeca.png', caption='Rebeca Abreu')
    with col2:
        st.markdown(":orange[**Scrum Master**]")
        st.image('joao.png', caption='João Pedro')
    with col3:
        st.markdown(":orange[**Team Member**]")
        st.image('laine.png', caption='Elaine Gonçalves')
    with col4:
        st.markdown(":orange[**Team Member**]")
        st.image('jess.png', caption='Jessica Tinoco')
    with st.container():
        col1, col2, col3, col4 = st.columns(4)
        with col1:
            st.markdown(":orange[**Team Member**]")
            st.image('joy.png', caption='Joyce Prudêncio')
        with col2:
            st.markdown(":orange[**Team Member**]")
            st.image('avila.png', caption='Vitor Ávila')
        with col3:
            st.markdown(":orange[**Team Member**]")
            st.image('hugo.png', caption='Vitor Hugo')
        with col4:
            st.markdown("")
```

Acesso a página de perguntas e chamada da função para perguntas

```
elif selected == "Perguntar":
    perguntar()
```

Acesso a página de BI e inclusão de imagem

```
elif selected == "Power BI":
    st.header(":orange[Visualize os indicadores do projeto!]", divider='orange')
    st.write("")
    st.write("")
    st.subheader("QR Code")
    st.image('qrcode_bi.png', width=300)
```

Acesso a página de Github e inclusão de imagem

```
elif selected == "GitHub":
    st.header(":orange[Conheça nossa documentação!]", divider='orange')
    st.write("")
    st.write("")
    st.subheader("QR Code")
    st.image('qrcode_git.png', width=300)
```

RAG

Importando as bibliotecas:

```
import pandas as pd
import pymysql

# Preparando o RAG
from langchain_community.utilities import SQLDatabase
from langchain.llms import OpenAI
from langchain_experimental.sql import SQLDatabaseChain
from langchain.prompts import PromptTemplate
from langchain.prompts.chat import HumanMessagePromptTemplate
from langchain.chat_models import ChatOpenAI
from langchain.schema import HumanMessage, SystemMessage
```

Importando chave API GPT:

```
key_gpt=" <KEY API GPT> "
llm = ChatOpenAI(temperature=0, openai_api_key=key_gpt, model="gpt-4")
```

Acessando base local SQL:

```
host='localhost'
username='root'
password='*****'
port='3306'
database='database_api3'
# construindo uma URI de conexão para o MySQL
mysql_uri = f"mysql+pymysql://{username}:{password}@{host}:{port}/{database}"
db = SQLDatabase.from_uri(mysql_uri, sample_rows_in_table_info=2)
db_chain = SQLDatabaseChain.from_llm(llm, db, verbose=True)
```

Definindo modo de resposta:

```
def retrieve_from_db(query: str) -> str:
    db_context = db_chain(query)
    db_context = db_context['result'].strip()
    return db_context
```

Definindo prompt para RAG

```
def generate(query: str) -> str:
    db_context = retrieve_from_db(query)

    system_message = """
    Você é um software analista de dados especializado em logística. Sua tarefa é
    responder às perguntas feitas pelos usuários
    fornecendo informações úteis para suas tomadas de decisão. As respostas devem
    ser concisas e diretamente relacionadas às informações
    fornecidas no contexto.

    Exemplo:

    Input:
    De quais municípios as fábricas enviam produtos?

    Context:
    Existem três municípios com fábricas que enviam produtos para os clientes. São
    eles:
    1. ITU
    2. ARARAQUARA
    3. JACAREÍ

    Output:
    As fábricas enviam produtos de três municípios: ITU, ARARAQUARA e JACAREÍ.

    Como um software analista de dados especializado em logística, é importante
    fornecer respostas precisas e consistentes em relação aos
    custos de frete. Sempre que uma pergunta sobre custo for feita, as seguintes
    convenções devem ser seguidas:
    1. Sempre utilize a moeda brasileira (R$).
    2. Utilize pontos como separadores de milhar e vírgulas como separadores
    decimais.
    3. Inclua o prefixo "R$" para indicar o valor monetário.
    4. Certifique-se de que a resposta seja clara e direta.

    Exemplo:

    Input:
    Qual é o meu custo total de frete?

    Context:
    O custo total de frete para o seu pedido é de R$ 10000,00.

    Output:
    O custo total do seu frete é R$ 10.000,00.

    Se o seu contexto conter um erro SQL retorne a mensagem: "Não sou capaz de
    responder essa pergunta"

    """

    human_qry_template = HumanMessagePromptTemplate.from_template(
        """Input:
        {human_input}

        Context:
        {db_context}

        Output:
        """
    )
```

```
messages = [  
    SystemMessage(content=system_message),  
    human_qry_template.format(human_input=query, db_context=db_context)  
]  
response = llm(messages).content  
return response
```