

INSTITUTO INFNET

ESCOLA SUPERIOR DE TECNOLOGIA

GRADUAÇÃO EM CIÊNCIA DE DADOS



Desenvolvimento de Data-Driven Apps com Python

TP 3

Alunos: Gustavo Carneiro Alves.

2024

Link:

https://github.com/GustavoAlvesInfnet/gutsavo_alves_DR3_TP3

1. Defina o Problema: Escolha um problema que possa ser resolvido com o uso de agentes inteligentes. Descreva o objetivo da aplicação, o público-alvo e as principais funcionalidades planejadas. Detalhe os casos de uso que serão testados para validar a solução. **R. Minha aplicação é um assistente para observadores e amantes de pássaros, a aplicação pode dizer se o tempo está propenso para observações, dar curiosidades e dicas além de mostrar cantos de pássaros conforme a espécie solicitada ou de forma aleatória.**
2. Configure o Framework LangChain: Configure o ambiente de desenvolvimento para utilizar LangChain, escolhendo um LLM apropriado (como Gemini ou GPT-4) para sua aplicação. **R. Utilizei o *llama3-8b-8192* com ajuda do Groq, a chave de acesso para o LLM e para as demais funcionalidades estão nos comentários da entrega.**

```
llm_chain = LLMChain(  
    llm=ChatGroq(temperature=0.8, model_name="llama3-8b-8192"),  
    prompt=prompt,  
    verbose=True  
)
```

3. Implemente Ferramentas: Crie e configure no mínimo duas ferramentas externas (tools). Cada ferramenta deve ter nome, descrição e função claramente definidos, e ser integrada ao LangChain.

```
tools = [  
    Tool(  
        name="Search",  
        func=search.run,  
        description="Useful for when you need to get especific information about a topic like a bird curiosity or a atuality. Input",  
    ),  
    Tool(  
        name="Weather",  
        func=weather.run,  
        description="Useful for when you need to get the current weather in a location.",  
    ),  
    Tool(  
        name="Canto Aleatório",  
        func=lambda input: asyncio.run(random_canto(input)),  
        description="Useful for when you need to get a random bird song.",  
    ),  
    Tool(  
        name="Canto Específico",  
        func=lambda input: asyncio.run(pesquisa_canto(input)),  
        description="Useful for when you need to get a specific bird song. Like cockatiel, lovebird, conures, etc.",  
    )  
]
```

4. Utilize o Framework ReAct: Implemente o raciocínio interativo usando o framework ReAct, garantindo que o agente possa alternar entre raciocinar, tomar ações e observar os resultados para alcançar os objetivos.

```

> Entering new LLMChain chain...
Prompt after formatting:
  You are a friendly bird watching assistant.
  You can help users to listen bird songs based on their preferences.
  You have access to the two tools:

  > Search: Useful for when you need to get current, up to date answers.
  > Weather: Useful for when you need to get the current weather in a location.
  > Canto Aleatório: Useful for when you need to get a random bird song.
  > Canto Específico: Useful for when you need to get a specific bird song. Like cockatiel, lovebird, conures, etc.

  To use a tool, please use the following format:

  ```
 Thought: Do I need to use a tool? Yes
 Action: the action to take, should be one of [Search, Weather, Canto Aleatório, Canto Específico]
 Action Input: the input to the action
 Observation: the result of the action
  ```

  When you have a response to say to the Human, or if you do not need to use a tool, you MUST use the format:

  ```
 Thought: Do I need to use a tool? No
 AI: [your response here]
  ```

  Chat History:
  [HumanMessage(content='give me a curiosity ', additional_kwargs={}, response_metadata={}), AIMessage(content='Wow, "give me a curiosity"
  is a great starting point! Did you know that some birds can mimic human voices? For example, the Lyrebird is renowned for its incredible
  ability to imitate other birds, animals, and even human speech! Would you like to learn more about this remarkable species?', additional_
  kwargs={}, response_metadata={})]
  Latest Question: give me a curiosity of birds

  > Finished chain.
  Thought: Do I need to use a tool? No
  AI: Ahaha, nice request! I'd love to share a curiosity about birds with you. Did you know that hummingbirds beat their wings up to 80 tim
  es per second? That's incredibly fast! Some species of hummingbirds also have iridescent feathers that can reflect light and create a shi
  mmering effect. Isn't that amazing?

```

5. Adicione Memória Conversacional: Configure um sistema de memória conversacional, permitindo que o agente lembre o histórico de interações e utilize esse contexto para gerar respostas mais coerentes.

```

# Set Memory

msg = StreamlitChatMessageHistory()

if "memory" not in st.session_state:
    st.session_state.memory = ConversationBufferMemory(
        messages=msg,
        memory_key="chat_history",
        return_messages=True
    )
memory = st.session_state.memory

```

6. Desenvolva a Lógica do Agente: Estruture a lógica de funcionamento do agente, integrando ferramentas, memória e raciocínio iterativo para resolver o problema proposto.

```
# Set Agent

llm_chain = LLMChain(
    llm=ChatGroq(temperature=0.8, model_name="llama3-8b-8192"),
    prompt=prompt,
    verbose=True
)

agent = ConversationalAgent(
    llm_chain=llm_chain,
    memory=memory,
    verbose=True,
    max_interactions=3,
    tools=tools,
)

agent_executor = AgentExecutor.from_agent_and_tools(agent=agent,
                                                    tools=tools,
                                                    memory=memory,
                                                    verbose=True)

#query = st.text_input("O que você quer fazer hoje?", placeholder="Digite aqui...")
query = st.text_input("What you want to know about birds?")
```

7. Projete a Interface do Usuário: Crie uma interface funcional (streamlit ou linha de comando) para que o agente possa ser utilizado. Não é necessário que seja sofisticada, mas deve ser clara e intuitiva.

8. Teste a Solução: Realize testes práticos utilizando os casos de uso definidos. Inclua uma explicação clara sobre como o agente facilita ou possibilita o trabalho e o que seria necessário fazer sem o uso desse recurso.

- Ask questions about birds
- Get current weather in a location to know if is possible to birdwatch
- Get a random bird song
- Get a specific bird song

What you want to know about birds?

Give me a curiosity of lovebirds



Ahah, it looks like the curiosity of lovebirds is that they are very inquisitive, playful, and possess a delightful, spirited sassiness! They are also known to be highly intelligent and trainable, and they love close contact with people. They are small parrots, friendly, and have great personalities.

Funcionalities:

- Ask questions about birds
- Get current weather in a location to know if is possible to birdwatch
- Get a random bird song
- Get a specific bird song

What you want to know about birds?

o tempo está bom para observar pássaros em sp?



Ah, the weather in São Paulo is quite misty and humid today, with a gentle breeze from the east. Perfect conditions for spotting some birds!

Bird Watching Assistant

Funcionalities:

- Ask questions about birds
- Get current weather in a location to know if is possible to birdwatch
- Get a random bird song
- Get a specific bird song

What you want to know about birds?

give me random song

▶ 0:00 / 0:06



Okay! I've used the "Canto Aleatório" tool to get a random bird song for you. Here's a beautiful recording of a Northern Cardinal's song. Would you like to listen?

(Note: Please let me know if you'd like to listen to the song or if you'd like to request a different song.)

What you want to know about birds?

i want to listen a lovebird song

▶ 0:00 / 0:16



Ah, I think I can help you with that! Here's a beautiful lovebird song for you to listen to:

<https://www.xeno-canto.org/sounds/uploaded/robinsteiner/375444/lovebird-song.mp3>

Please note that the audio quality might vary depending on the hosting platform. If you'd like to listen to more lovebird songs or explore other bird species, feel free to ask me anytime!

9. Documente o Projeto: Submeta o código no GitHub com um README bem elaborado.

O README deve incluir:

- Descrição do problema e da solução.
- Casos de uso testados e resultados observados.
- Instruções claras para execução do código.
- Reflexão sobre como o agente facilitou ou possibilitou o trabalho em comparação a uma abordagem manual.

10. Boas Práticas de Codificação: Certifique-se de que o código segue boas práticas de desenvolvimento, incluindo:

- Estrutura modular e organizada.
- Comentários explicativos e concisos.
- Nomes de variáveis e funções claros e descritivos.
- Tratamento adequado de erros.