

5 - Leitura: Construindo um fluxo Langflow (II)

Vitor Manoel Emidio Machado

1. Nvidia une forças com Langflow para acelerar inteligência artificial personalizada

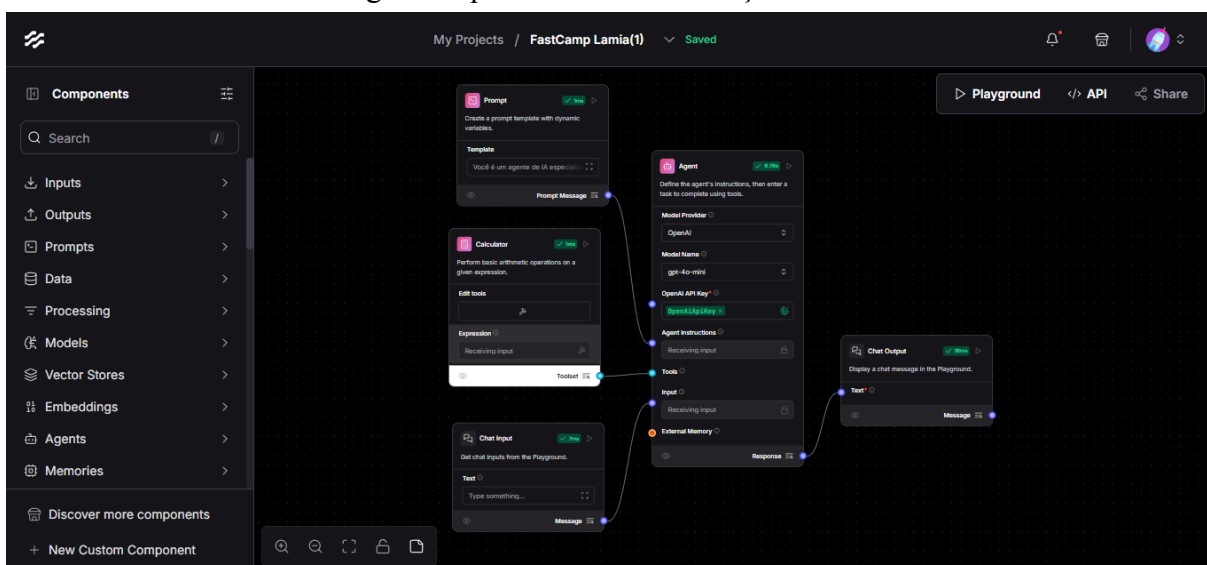
A grande empresa de tecnologias Nvidia, gerou uma parceria com a startup brasileira Langflow a fim de integrar a sua plataforma visual ao Projeto R2X, que visa oferecer assistentes de inteligência artificial personalizados, aprimorados pelo poder de GPUs RTX e tecnologias avançadas como RTX Neural Faces e NVIDIA Áudio 2Face™.

Essa parceria gera muitos benefícios ao usuário da plataforma Langflow, como: otimização do treino e implementação dos agentes IA, diminuição do tempo de resposta e democratização do acesso a IA, e além disso, temos a ascensão do Brasil dentro do mercado de tecnologia disruptivas.

Por parte da Nvidia, também teremos muitos benefícios proveniente dessa parceria, um dos mais interessantes seria para os usuários das placas de vídeo RTX, que poderão criar agentes personalizados diretos no Langflow, além da melhoria na qualidade da renderização de expressões faciais que imitam reações humanas, sincronização de áudio e vídeo dentre outras avanços. Trazendo o Langflow e o Brasil como protagonistas no cenário global de inovações tecnológicas.

2. A Beginners Guide to Build Agents in Langflow.

Neste guia, nos é explicado como criar o seu primeiro agent dentro do Langflow, para efeitos de teste eu criei um agente especializado na resolução de derivadas.

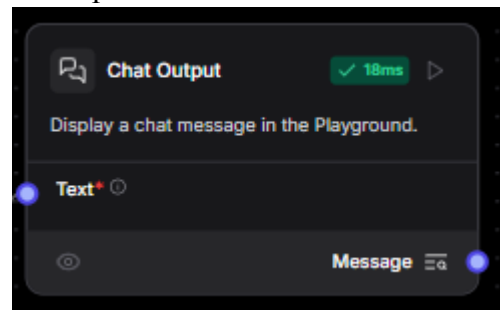
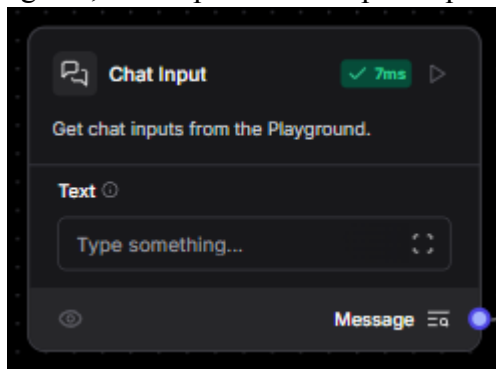


Nesse sentido, começando pelo agente em si, eu utilizei a Api da OpenAi como modelo, além disso dentro dele temos funções como:

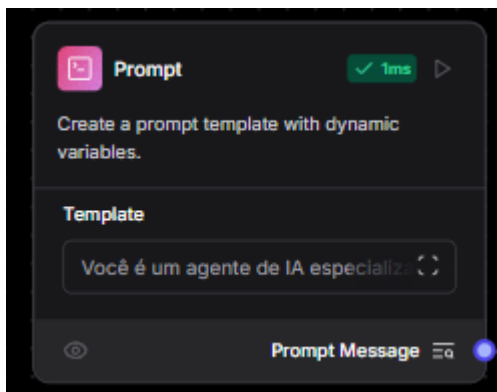
- **Tools** = local onde conectarmos as ferramentas necessárias para o funcionamento do agent.
- **Agent instructions** = local onde iremos instruir o comportamento do agente, no meu caso eu conectei a ela um prompt.
- **Input** = Será por onde nossa mensagem alcançará o agente e iniciará seu funcionamento.
- **External memory** = Aqui podemos adicionar uma memória externa ao agente.
- **Response** = Local no qual a resposta do agente irá sair

Agora, sobre as ferramentas do próprio Langflow que eu utilizei

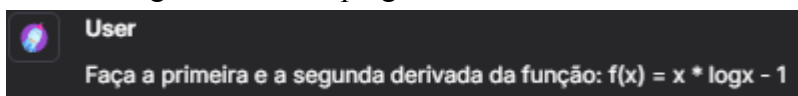
- **Chat Input e Output:** respectivamente, o input será utilizado para a ‘comunicação’ com o agente, e o output será a resposta que o agente criou para a resposta feita.



- **Prompt:** Local onde as instruções de como o agente deve se comportar são mantidas, no meu caso o meu prompt instrui o agente em resolver as derivadas que o indivíduo indagar sobre.



- Teste do agente: fiz uma pergunta sobre as duas derivadas da função: $f(x) = x * \log x - 1$



Ele conseguiu responder com exite, mesmo que a saída da resposta não esteja no melhor formato, mas bate com a resposta esperada ($f'(x) = \log x + 1$) ($f''(x) = 1/x$)

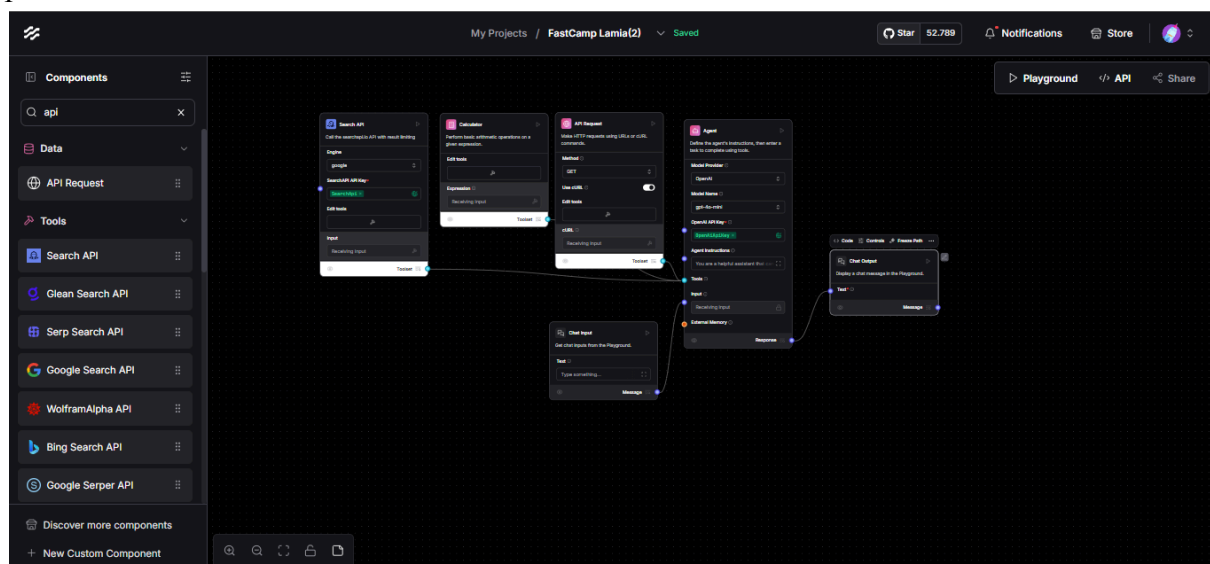
Resumo

- **Primeira Derivada:** ($f'(x) = \log(x) + 1$)
- **Segunda Derivada:** ($f''(x) = \frac{1}{x}$)


Se precisar de mais alguma coisa, é só avisar!

3. Create a problem-solving agent.


Para esse próximo conteúdo nos é apresentado uma forma de criar um agente que resolva problemas em geral, uma adição ao exemplo anterior, seria a 2 novas Tools, Search Api Tool, que no caso se trata de uma ferramenta que utiliza uma api que possibilita que o agente faça buscas/pesquisas na internet, e o Api Request, que como o próprio nome já diz, faz requisições de Api, nesse momento já temos uma Agent IA pronto para solucionar problemas.




Para testes, pedi para o agente personificar um excelente programador de Python, e fiz uma pergunta para ele responder.

**User**


You are a knowledgeable software developer who uses the tools at your disposal

**AI** gpt-4o-mini

**Finished**

✓ 2.0s

Thank you! I'm here to assist you with any software development questions or tasks you may have. Whether you need help with coding, debugging, or understanding concepts, feel free to ask!

**User**

In Phyton how can I import a csv file of beauty product sales, view its content and generate a graph?

Ele respondeu com êxito, gerando um código e explicando o seu funcionamento.

```
python

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Step 1: Import the CSV file
file_path = 'path/to/your/beauty_product_sales.csv' # Update with your file path
data = pd.read_csv(file_path)

# Step 2: View the content
print(data.head()) # Display the first 5 rows of the DataFrame

# Step 3: Generate a graph
# Example: Let's say you want to plot sales over time
# Ensure your CSV has a 'Date' and 'Sales' column
data['Date'] = pd.to_datetime(data['Date']) # Convert 'Date' column to datetime
plt.figure(figsize=(10, 6))
sns.lineplot(data=data, x='Date', y='Sales') # Adjust 'Sales' to your actual sales column
plt.title('Beauty Product Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Notes:

Dificuldades.

Não tive dificuldades com o tema pois já tenho um pouco de conhecimento com a plataforma, e por se tratar de LowCode se torna mais fácil.

Conclusão.

Nessa atividade foi possível colocar em prática os conhecimentos adquiridos até o momento na criação de um agente completo, diferente da prática em python, agora temos uma variedade muito maior de ferramentas à nossa disposição e o mais importante, de maneira fácil e simples, acredito que o maior empecilho seja obter as APIs porém depois disso se torna muito intuitivo modelar o agente, o LangFlow realmente facilita a vida de quem tem pouca experiência assim como eu.

Referências.

[Nvidia une forças com Langflow para acelerar inteligência artificial personalizada](#)
[Create a problem-solving agent | Langflow Documentation](#)
[A Beginner's Guide to Building Agents in Langflow | DataStax](#)
[Langflow overview](#)

