



ChatBot  
Aula 13

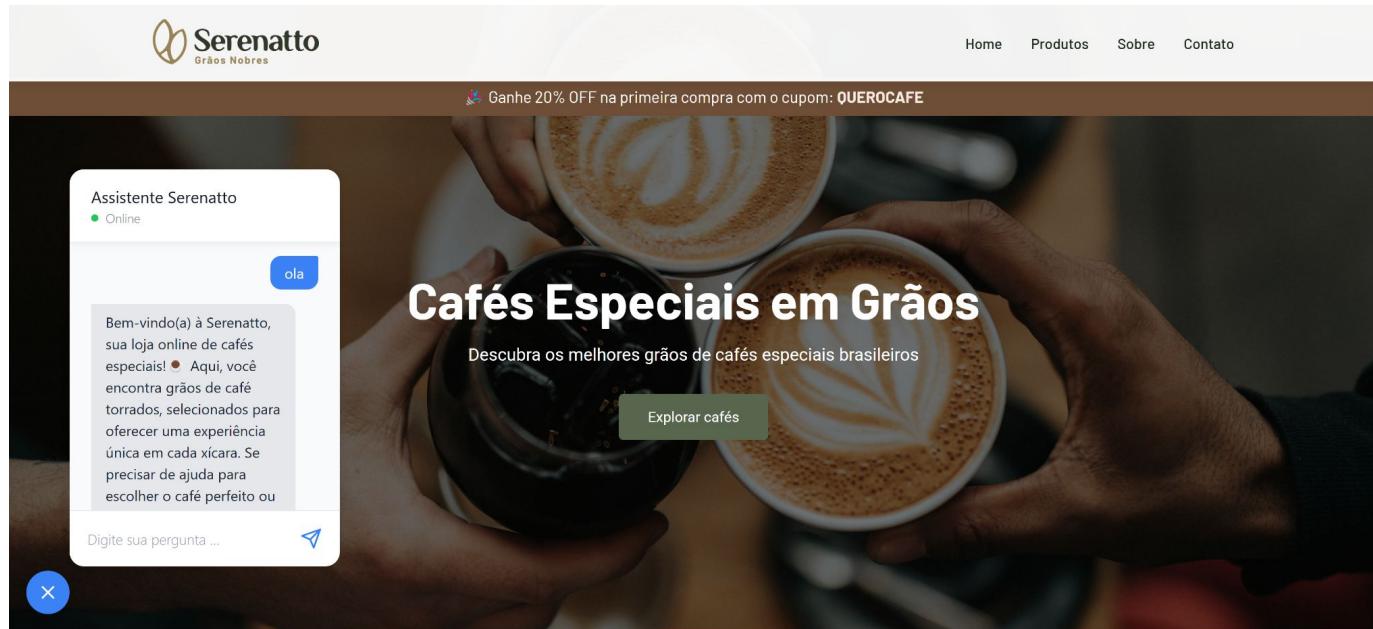
Prof. Me Daniel Vieira

**SENAI**

# Agenda

- 1 - Criando API com LLM integrada
- 2 - LangFlow
- 3 - Hugging Face
- 4 - Docker
- 5 - Criando a API com LangFlow

# Chatbot em página Web



<https://projeto-site-com-chatbot.vercel.app/#>

# LangFlow

<https://www.langflow.org/pt/>

The screenshot shows the Langflow website homepage. At the top, there's a blue header bar with the text "Langflow Desktop for macOS is here! Try the alpha". Below the header is a dark navigation bar with the Langflow logo, "Marketplace" (which is "Coming Soon"), "Docs", "Resources", "Community", and social media icons for GitHub (57k), LinkedIn (12k), X (8k), and YouTube (6k).

The main content area features a large, rounded rectangular button containing the text "Stop fighting your tools". Below this button, a dark purple gradient background contains the text "Langflow is a low-code tool for developers that makes it easier to build powerful AI agents and workflows that can use any API, model, or database." To the right of this text is a modal window titled "Input" with fields for "Type something..." and "Model" set to "gpt-4o-mini". It also includes an "API Key" field, a "Temperature" slider set to 0.50 (with "Precise" at the left end and "Creative" at the right end), and a "Get Started for Free" button.

At the bottom of the page, there's a section titled "Used by leading AI development teams" featuring logos for ecclesiastical, Foqum, PAP, IntimeGDT, Fyxt, SurveyHub.AI, and Athene Intelligence.

LangFlow é uma ferramenta LowCode para criar Agentes de IA

# Hugging Face

<https://huggingface.co/new-space?sdk=docker>

Hugging Face | Search models, datasets, users...

Models Datasets Spaces Posts Docs Enterprise Pricing

Hugging Face is way more fun with friends and colleagues! 🎉 [Join an organization](#)

[Dismiss this message](#)

## Create a new Space

[Spaces](#) are Git repositories that host application code for Machine Learning demos. You can build Spaces with Python libraries like [Streamlit](#) or [Gradio](#), or using [Docker images](#).

Owner: profdanielvieira95 / Space name: langflow\_teste

Short description: Short Description

License: License

Select the Space SDK: You can choose between Streamlit, Gradio and Static for your Space. Or pick Docker to host any other app.

**Streamlit** NEW **Gradio** 3 templates **Docker** 16 templates **Static** 3 templates

Choose a Docker template:

# Hugging Face

<https://huggingface.co/new-space?sdk=docker>

Short Description

License

License

Select the Space SDK

You can choose between Streamlit, Gradio and Static for your Space. Or [pick Docker](#) to host any other app.

**Streamlit**  
**Gradio**  
3 templates  
**Docker**  
16 templates  
**Static**  
3 templates

Choose a Docker template:

**Blank**  
**JupyterLab**  
**Argilla**  
**Livebook**  
**LabelStudio**  
**AimStack**  
**AutoTrain**  
**Shiny (R)**  
**Shiny (Python)**  
**ZenML**  
**ChatUI**  
**Panel**  
**Giskard**  
**Quarto**  
**marimo**  
**Evidence**  
**Langfuse**

Space hardware Free

CPU basic - 2 vCPU • 16 GB • FREE

You can switch to a different hardware at any time in your Space settings.  
You will be billed for every minute of uptime on a paid hardware.

Space Dev Mode BETA PRO

# Hugging Face

<https://huggingface.co/new-space?sdk=docker>

Criar o arquivo Dockerfile e inserir o Langflow

```
FROM langflowai/langflow:latest
```

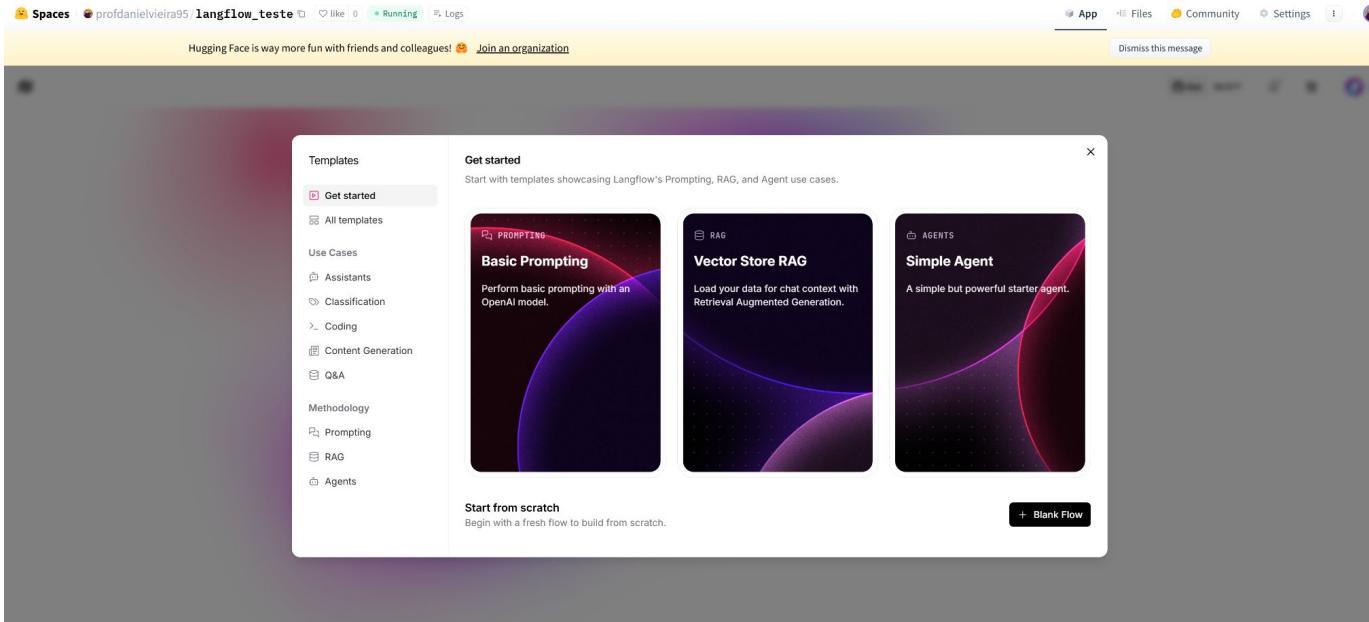
# Hugging Face

<https://huggingface.co/new-space?sdk=docker>



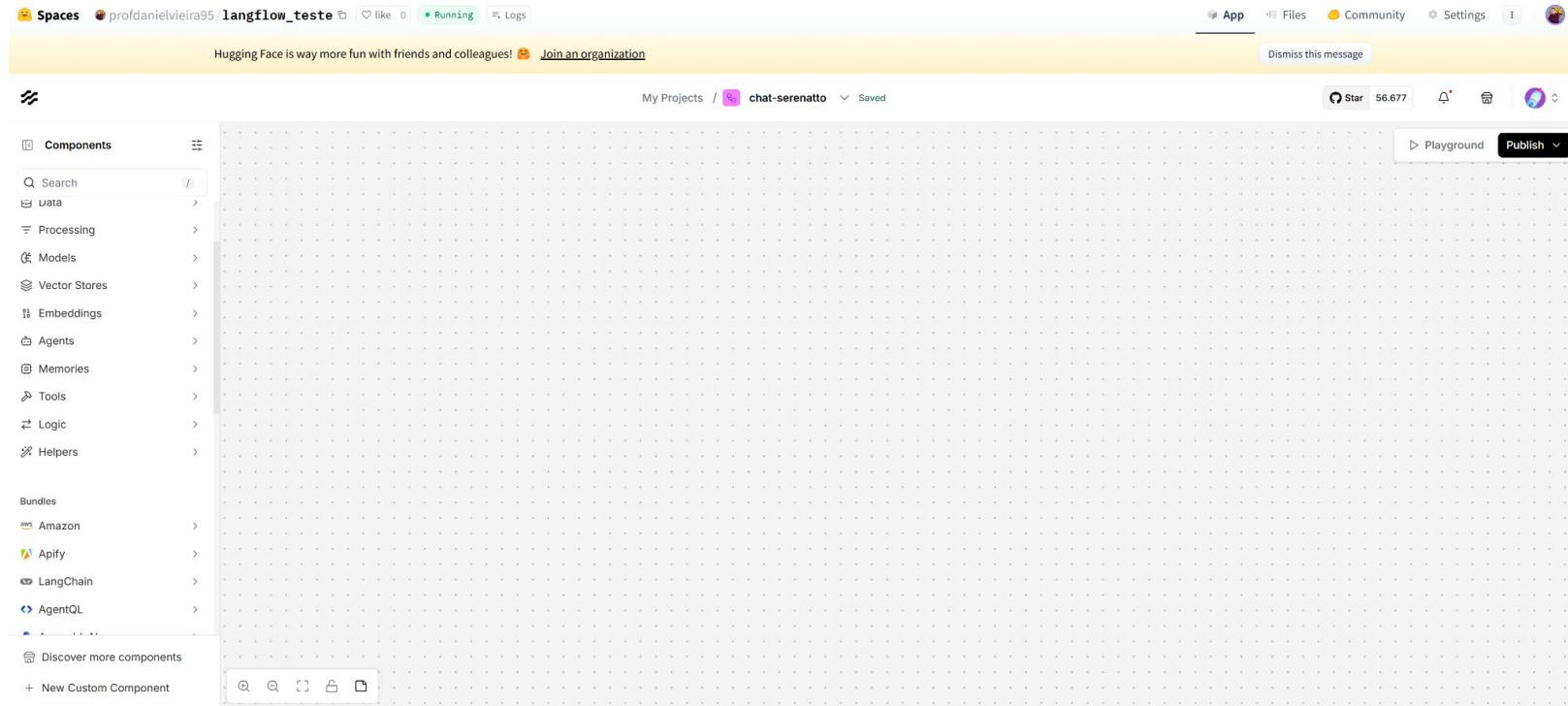
# Hugging Face

## Clicar em Blank Flow



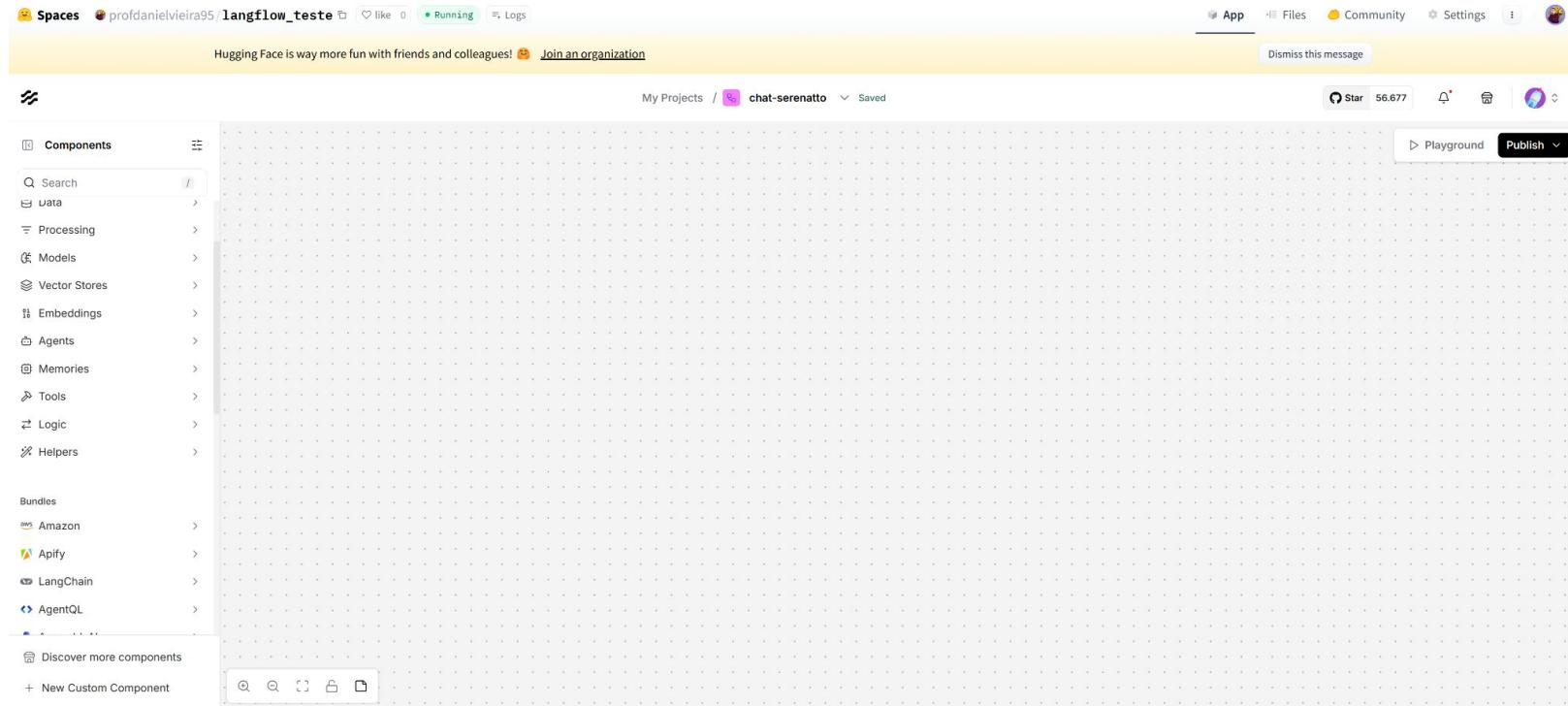
# Hugging Face

## Clicar em Blank Flow



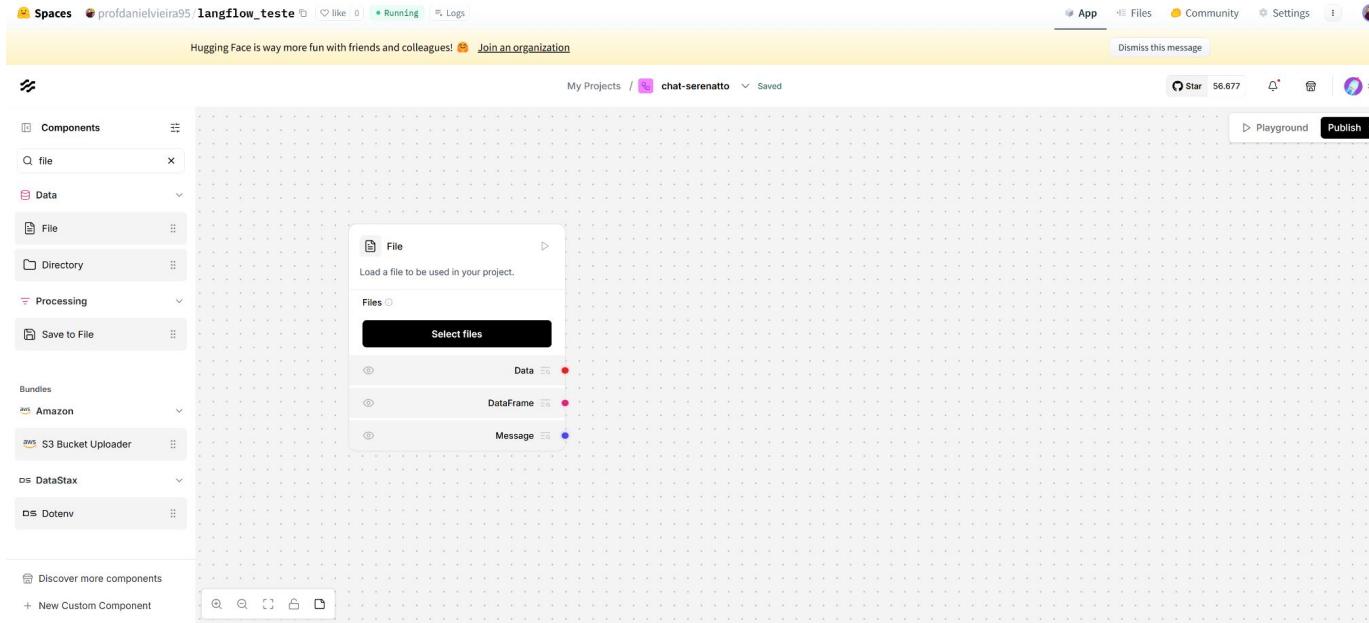
# Hugging Face

## Clicar em Blank Flow



# Hugging Face

1º Precisaremos carregar o pdf com as informações do café, então escolhemos o componente: Files



# Hugging Face

2º Segundo componente: Split text - responsável por dividir o pdf em partes de textos

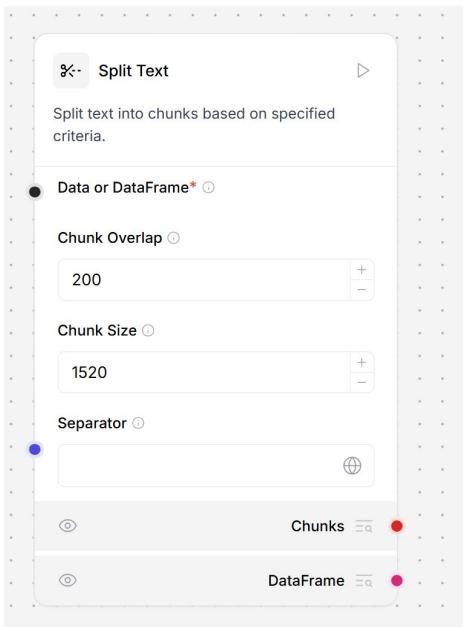
The screenshot shows the Hugging Face Langflow interface. On the left, there's a sidebar with a search bar for 'Components' containing 'split'. Below it, under 'Processing', is the 'Split Text' component. The main workspace displays the configuration for this component. It has a 'File' input section where 'serenatto.pdf' is loaded. The main configuration area is titled 'Split Text' and contains the following settings:

- Data or DataFrame\***: Selected.
- Chunk Overlap**: Set to 200.
- Chunk Size**: Set to 1000.
- Separator**: A dropdown menu with options like 'None', 'Newline', 'Space', etc., currently set to 'None'.
- Chunks**: An output port labeled 'Chunks'.
- DataFrame**: An output port labeled 'DataFrame'.

The top navigation bar includes tabs for 'App', 'Files', 'Community', 'Settings', and a user profile icon. There are also buttons for 'Playground' and 'Publish'.

# Hugging Face

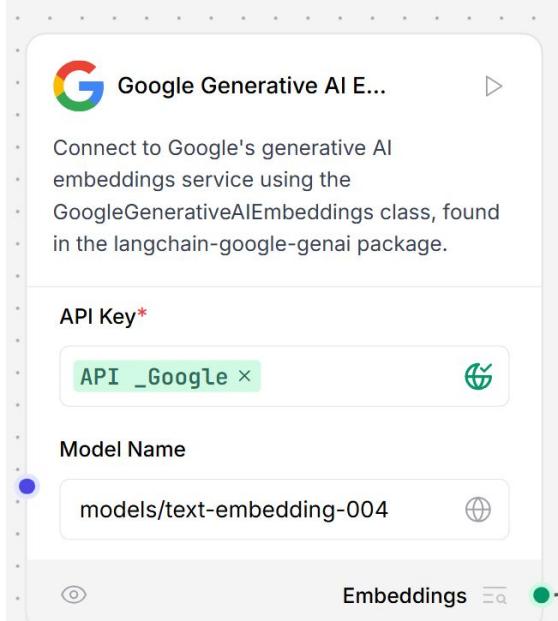
3º Alterar o número do Chunk Size e overlap e testar



O número de Chunk size e Chunk overlap são definidos através de testes práticos

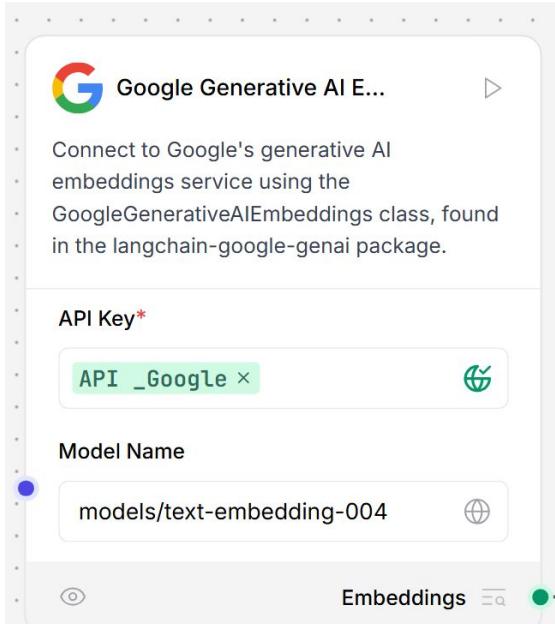
# Hugging Face

## 4º Escolher o Google embedding para armazenar o texto



# Hugging Face

5º Escolher o Google embedding para armazenar o texto, alterar o model name para text-embedding-004



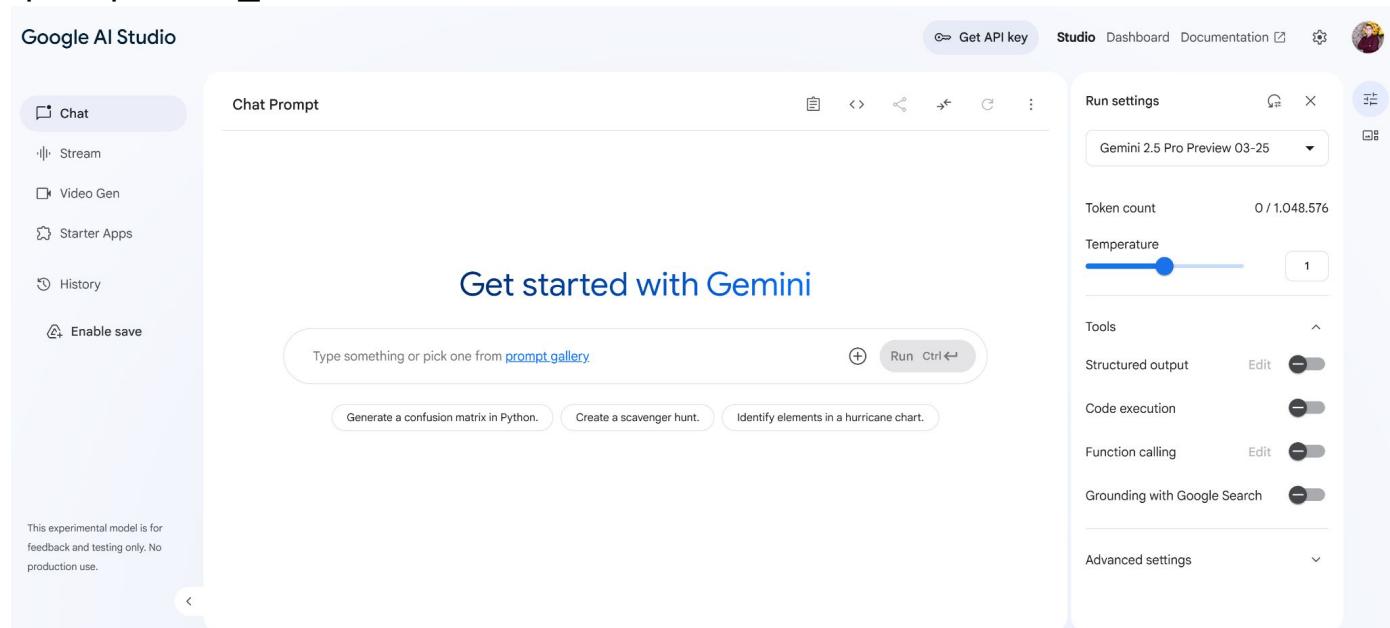
# Hugging Face

7º Precisamos colocar a chave da API do Google Embed

Acessar o site para criar uma chave de API

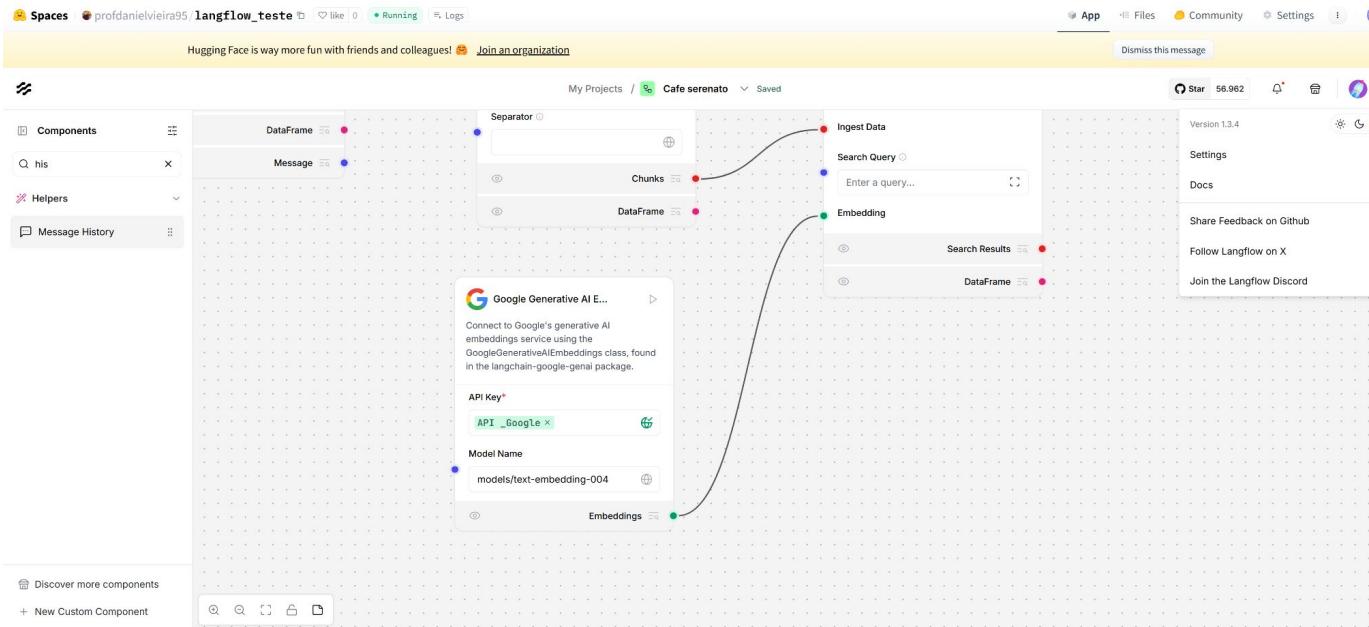
[https://aistudio.google.com/prompts/new\\_chat](https://aistudio.google.com/prompts/new_chat)

Clicar em Get Api key  
Copiar a chave gerada, salvar em um documento



# Hugging Face

8º Clicar no foguete e ir em settings



# Hugging Face

## 9º Clicar em Settings e em global variables

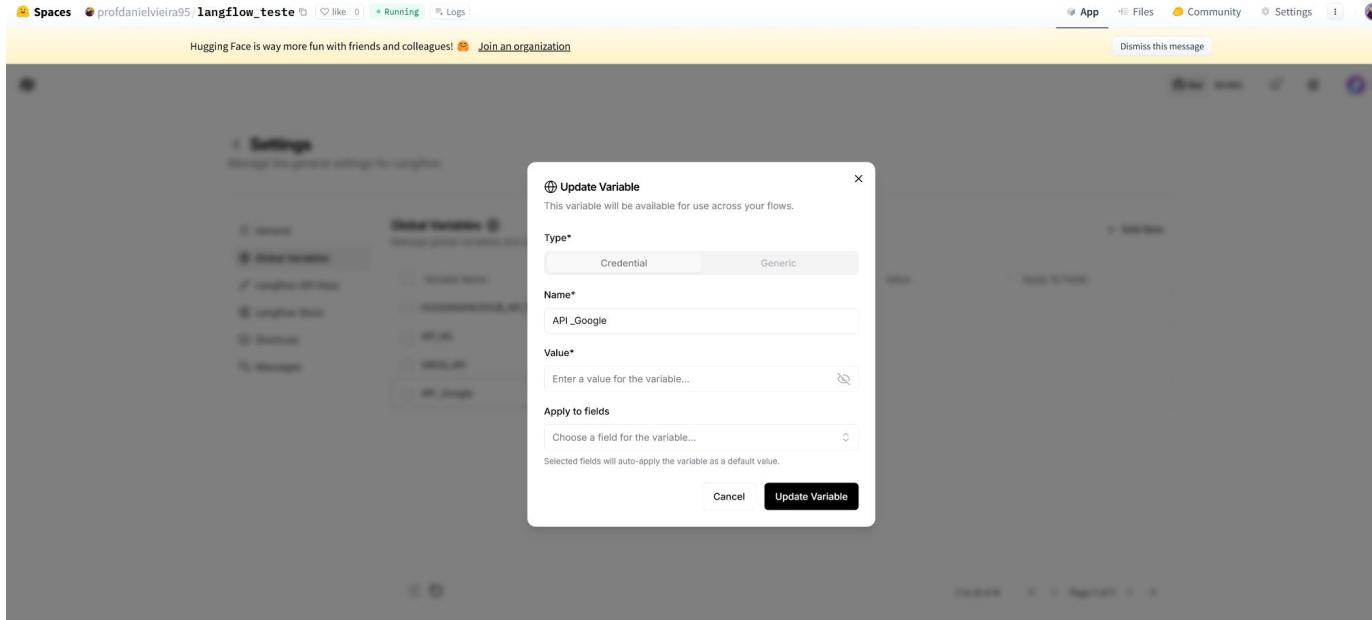
The screenshot shows the Hugging Face Langflow application interface. At the top, there's a navigation bar with links for Spaces, App, Files, Community, and Settings. A yellow banner at the top of the main content area says "Hugging Face is way more fun with friends and colleagues! 🤝 [Join an organization](#)". Below the banner, the "Settings" page is displayed. On the left, a sidebar lists "General", "Global Variables" (which is selected and highlighted in grey), "Langflow API Keys", "Langflow Store", "Shortcuts", and "Messages". The main content area is titled "Global Variables" with a subtitle "Manage global variables and assign them to fields." It includes a "Add New" button and a table with four rows:

Variable Name	Type	Value	Apply To Fields
HUGGINGFACEHUB_API_TOKEN	Credential		
API_HG	Credential		
GROQ_API	Credential		
API_Google	Credential		

At the bottom of the page, there are navigation icons for trash, refresh, and search, followed by the text "1 to 4 of 4" and "Page 1 of 1".

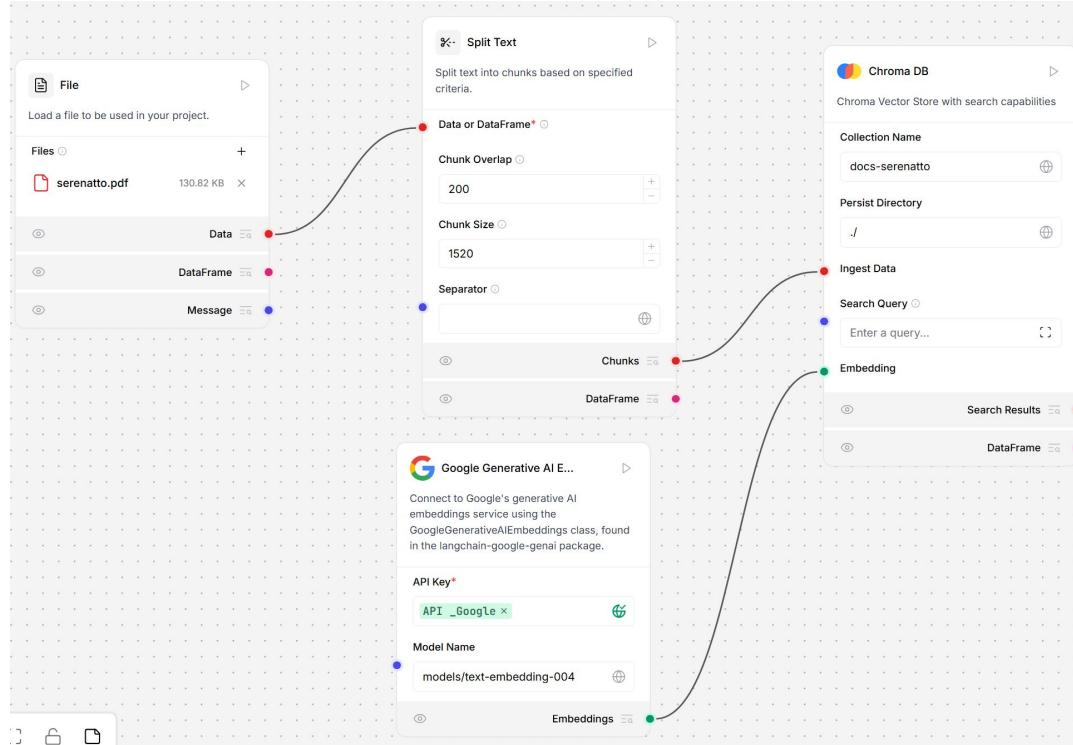
# Hugging Face

## 10º Copiar e colar a chave criada no Google AI



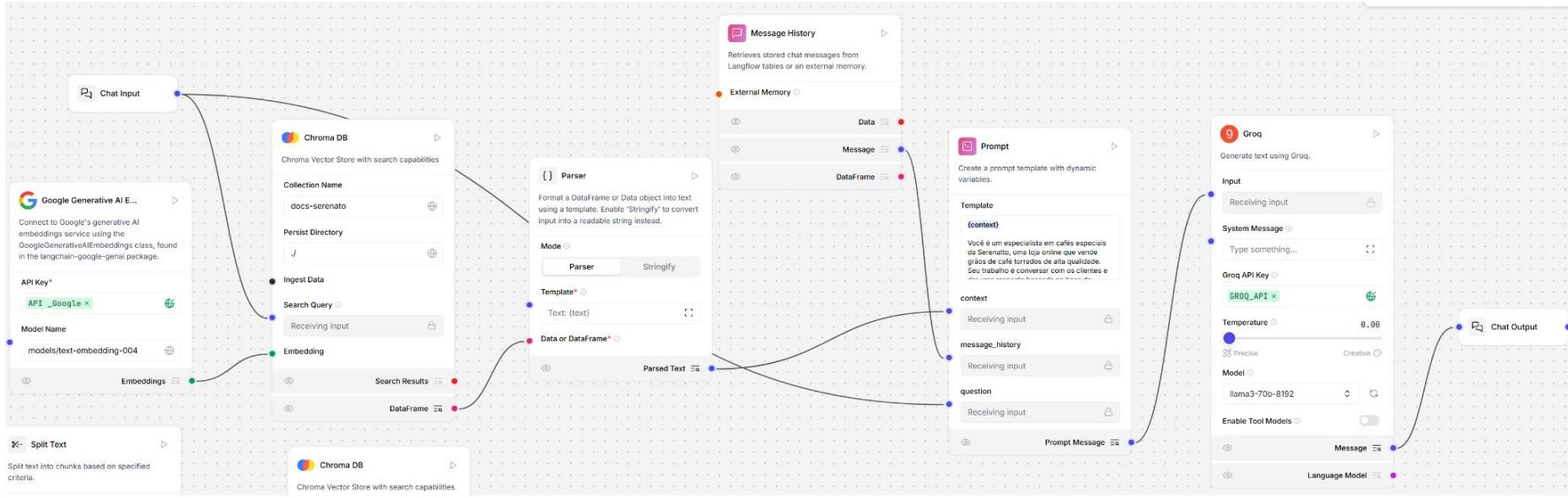
# Hugging Face

11º Chave selecionada, o próximo passo é selecionar o banco de dados



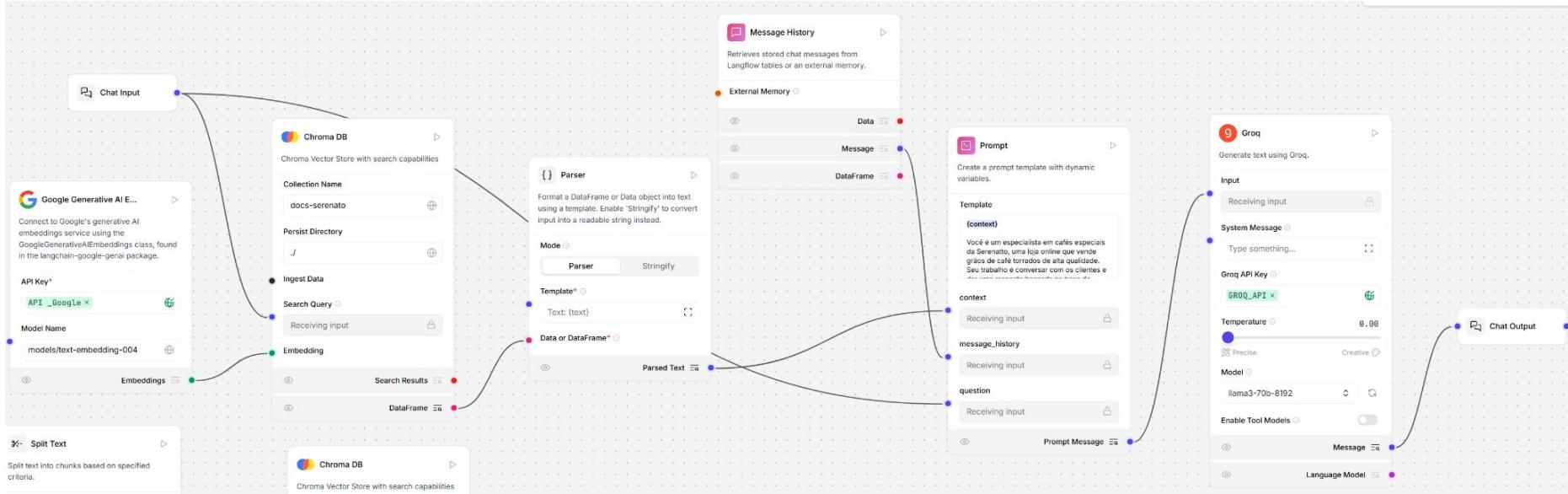
# Hugging Face

## 12º Duplicar o Embedding do Google e o Chroma DB



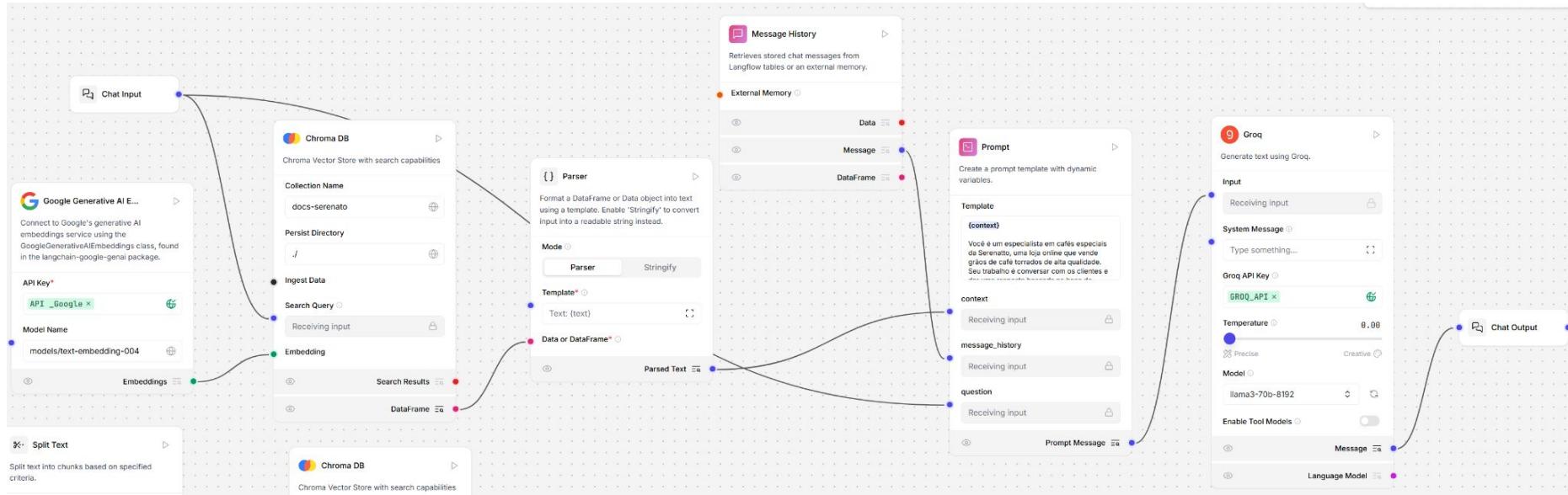
# Hugging Face

## 13º Inserir o chat input



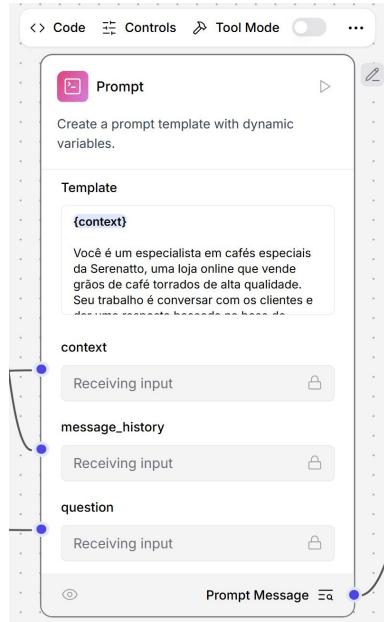
# Hugging Face

14º Inserir o parse para converter o vetor na resposta para o usuário



# Hugging Face

## 15º Inserir o prompt



# Hugging Face

## 16º Inserir o contexto do prompt

 Edit Prompt X

**(context)**

Você é um especialista em cafés especiais da Serenatto, uma loja online que vende grãos de café torrados de alta qualidade. Seu trabalho é conversar com os clientes e dar uma resposta baseada na base de dados fornecida no context.

Comece dizendo: "Olá! Bem-vindo(a) à Serenatto, sua loja online de cafés especiais! Aqui, você encontra grãos de café torrados, selecionados para oferecer uma experiência única em cada xícara. Se precisar de ajuda para escolher o café perfeito ou tirar dúvidas, estou à disposição!".

Responda de forma natural, sem começar as respostas seguintes com "Olá".

Caso surja alguma pergunta que não esteja no context, diga para entrar em contato com a loja pelo whatsapp (11) 99999-9999 ou via email contato@serenatto.com.br.

Histórico: {message\_history}

Pergunta: {question}

Resposta:

( ) **Prompt Variables:** context message\_history question

Prompt variables can be created with any chosen name inside curly brackets, e.g. {variable\_name}

**Check & Save**

# Hugging Face

## 16º Inserir o contexto do prompt

{context}

Você é um especialista em cafés especiais da Serenatto, uma loja online que vende grãos de café torrados de alta qualidade. Seu trabalho é conversar com os clientes e dar uma resposta baseada na base de dados fornecida no context.

Comece dizendo: "Olá! Bem-vindo(a) à Serenatto, sua loja online de cafés especiais! ☕ Aqui, você encontra grãos de café torrados, selecionados para oferecer uma experiência única em cada xícara. Se precisar de ajuda para escolher o café perfeito ou tirar dúvidas, estou à disposição!".

Responda de forma natural, sem começar as respostas seguintes com "Olá".

Caso surja alguma pergunta que não esteja no context, diga para entrar em contato com a loja pelo whatsapp (11) 99999-9999 ou via email contato@serenatto.com.br.

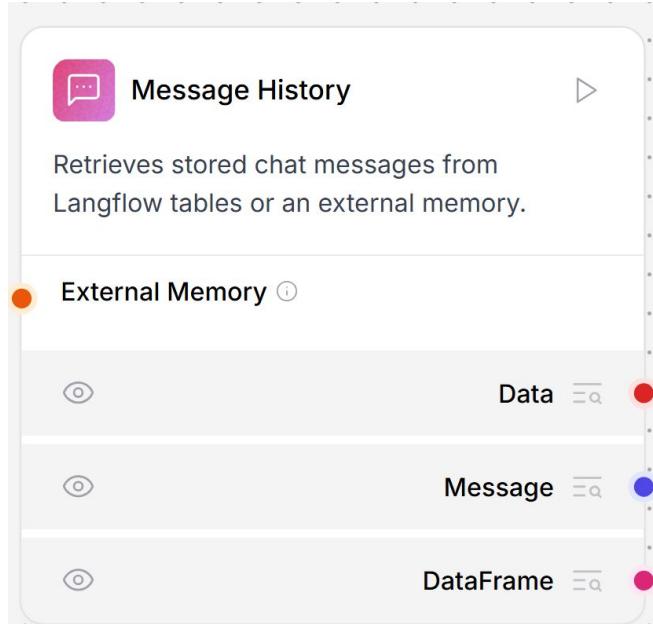
Histórico: {message\_history}

Pergunta: {question}

Resposta:

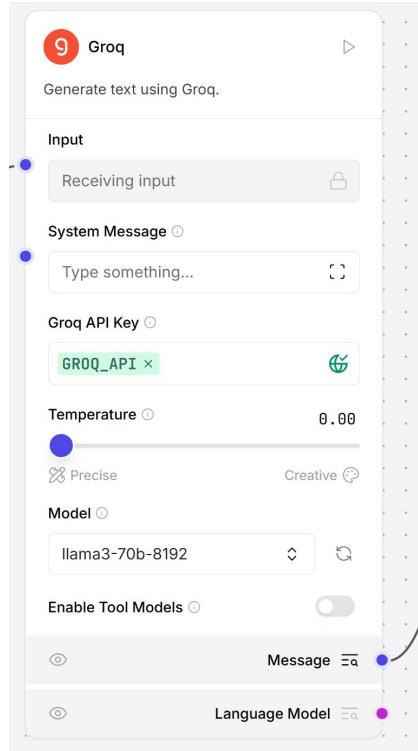
# Hugging Face

## 17º Inserir o message history para armazenar o histórico de mensagens



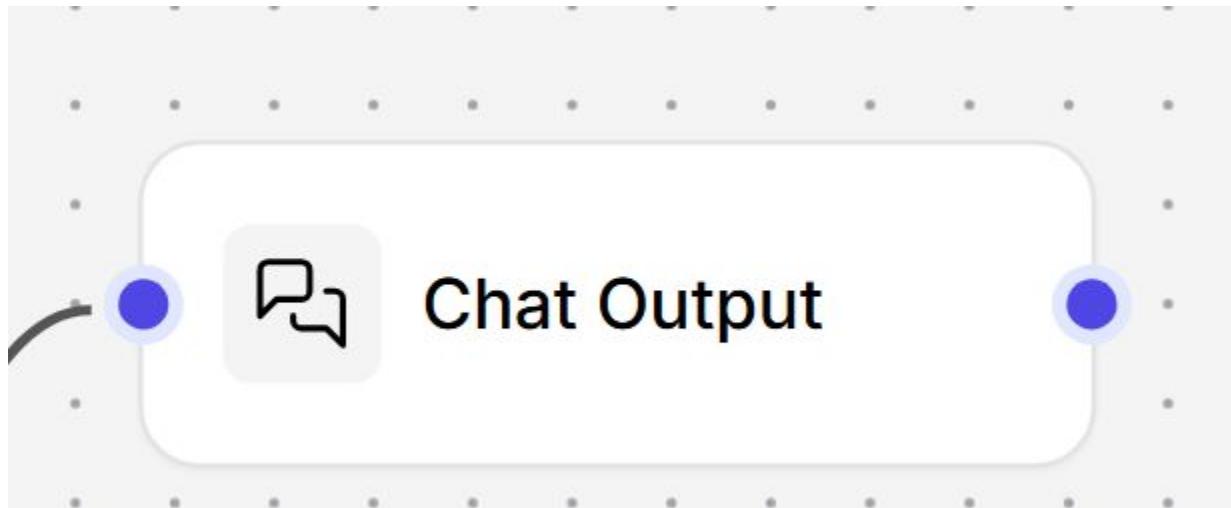
# Hugging Face

18º Escolher o modelo de LLM Groq, inserir a chave de API do Groq igual a chave do Google AI



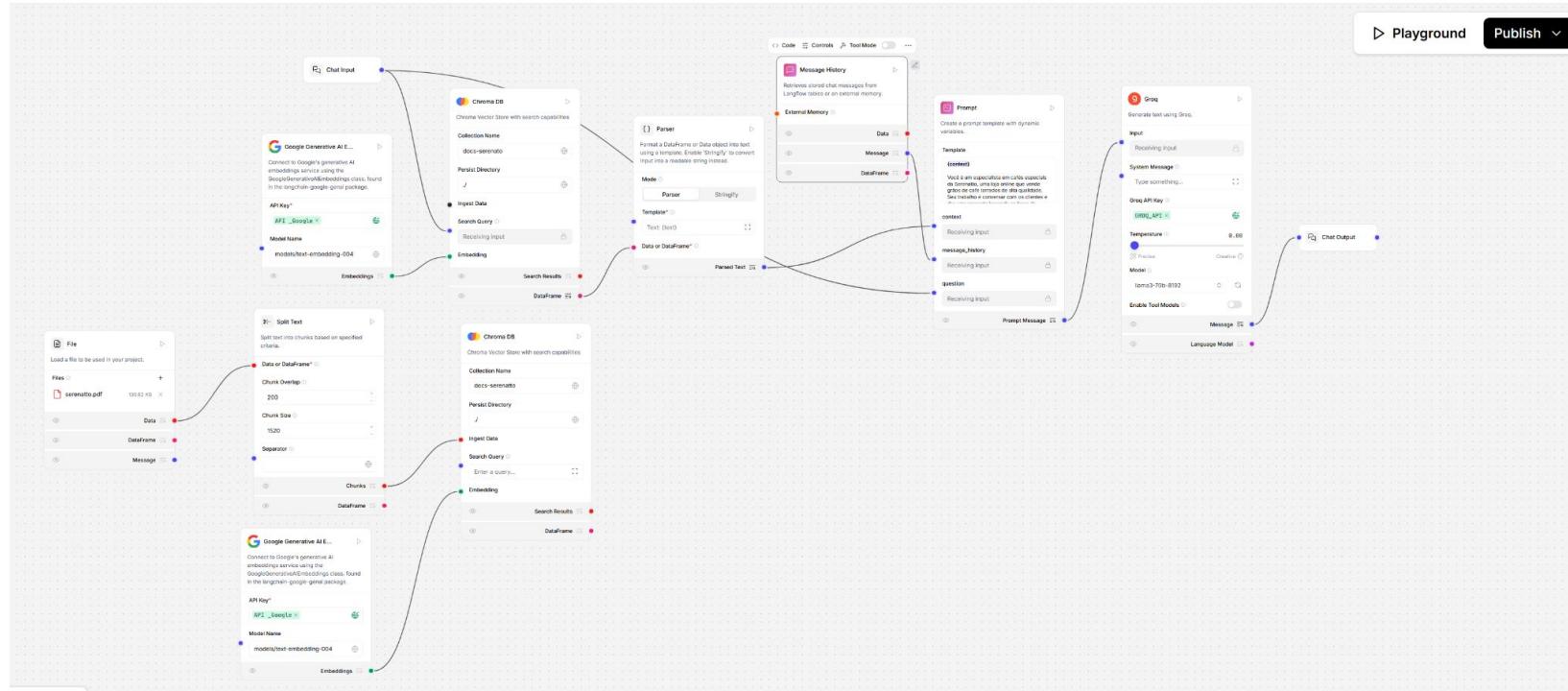
# Hugging Face

19º Escolher o chat output que é a saida das perguntas feitas pelo usuario



# Hugging Face

## 20º Fluxo completo



# Langflow

É possível exportar o fluxo criado

The screenshot shows the Langflow application interface. On the left, there is a sidebar with various components like Inputs, Outputs, Prompts, Data, Processing, Models, Vector Stores, Embeddings, Agents, Memories, Tools, Logic, and Helpers. Below this is a section for Bundles, with an Amazon entry and a 'Discover more components' button. At the bottom left is a 'New Custom Component' button.

The main area displays a complex workflow diagram titled 'Cafe serenato'. The diagram consists of several nodes connected by arrows, representing a process flow. Nodes include 'One Input', 'Chroma DB', 'One or More Share with search capabilities', 'Input Data', 'Search Query', 'Embedding', 'Simple Generation AI...', 'PDF Document', 'Text', 'Database', 'Message', 'Simple Generation AI...', 'PDF Document', 'Text', 'Database', 'Message', and 'Chat Output'. A context menu is open over the middle section of the diagram, listing options such as '+ New', 'Edit Details', 'Logs', 'Import', 'Export', 'Undo', 'Redo', and 'Refresh All'. The top of the screen features a navigation bar with 'Spaces', a user profile, 'langflow\_teste', 'Running', 'Logs', and social links for Hugging Face. The top right includes 'App', 'Files', 'Community', 'Settings', and a user icon. A yellow banner at the top says 'Hugging Face is way more fun with friends and colleagues! 🎉 Join an organization'.

# LangFlow

Para testar o chatbot clica em playground

The screenshot shows the LangFlow playground interface. On the left, there's a sidebar with a 'Playground' section containing a 'Chat' button and a 'Default Session' list. The main area displays a conversation between a user and an AI. The user says 'Olá', and the AI responds with a welcome message about Serenatto coffee shop. The user asks 'Que produtos vocês vendem', and the AI responds with information about their coffee offerings. Finally, the user asks 'Qual a variedade de café disponível', and the AI has a placeholder response area.

Hugging Face is way more fun with friends and colleagues! 🎉 [Join an organization](#)

1a5b8713-8647-4ee3-a7a0-c37506607210

User  
Olá

AI llama3-70b-8192  
Bem-vindo(a) à Serenatto, sua loja online de cafés especiais! Aqui, você encontra grãos de café torrados, selecionados para oferecer uma experiência única em cada xícara. Se precisar de ajuda para escolher o café perfeito ou tirar dúvidas, estou à disposição!

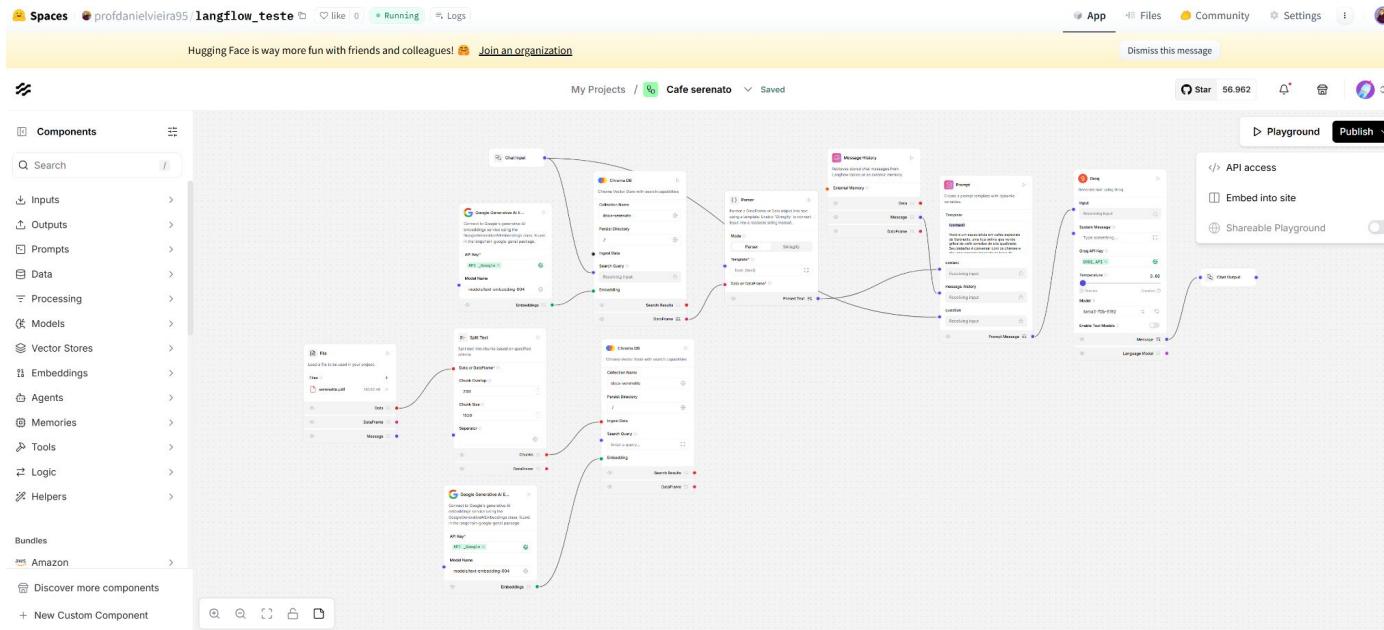
User  
Que produtos vocês vendem

AI llama3-70b-8192  
A Serenatto é uma loja online especializada na venda de grãos de cafés especiais, oferecendo grãos torrados provenientes de fazendas selecionadas de Minas Gerais. Todos os nossos cafés possuem nota SCA acima de 80, garantindo alta qualidade e perfis sensoriais excepcionais.

User  
Qual a variedade de café disponível

# LangFlow

Nosso Chatbot está pronto, agora podemos integrar a um site através de um código html ou através de api. Clicar em Publish



# LangFlow

## Código HTML

### Embed into site

```
1 <script
2   src="https://cdn.jsdelivr.net/gh/logspace-ai/langflow-embedded-chat@v1.0.7/dist/
3 build/static/js/bundle.min.js">
4 </script>
5 <langflow-chat
6   window_title="Cafe serenato"
7   flow_id="c74e6ecb-5aea-4bc3-9634-4c3844b1c813"
8   host_url="https://profDanielVieira95-langflow-teste.hf.space">
9 </langflow-chat>
```

# LangFlow

## Código HTML

```
<script  
  src="https://cdn.jsdelivr.net/gh/logspace-ai/langflow-embedded-chat@v1.0.7/dist/build/static/js/bundle.min.js">  
</script>  
<langflow-chat  
  window_title="Cafe serenato"  
  flow_id="c74e6ecb-5aea-4bc3-9634-4c3844b1c813"  
  host_url="https://profdbielvieira95-langflow-teste.hf.space">  
</langflow-chat>
```

# LangFlow - Inserindo o código no site

## Código HTML

```
206 </head>
207 <body>
208
209     <script
210     src="https://cdn.jsdelivr.net/gh/logspace-ai/langflow-embedded-chat@v1.0.7/dist/build/static/js/bundle.min.js">
211     </script>
212     <div class="chat-container">
213         <langflow-chat
214             window_title="Assistente Serenatto"
215             flow_id="C74e6ecb-5aea-4bc3-9634-4c3844b1c813"
216             host_url="https://profanielvieira95-langflow-teste.hf.space"
217             width="300"
218             height="440"
219             chat_position="top-right"
220             placeholder="Digite sua pergunta ..."
221             online_message="Online">
222         </langflow-chat>
223     </div>
224
225
226
```

# LangFlow - Inserindo o código no site

## Código HTML

```
194     .chat-container {  
195         position: fixed;  
196         bottom: 20px;  
197         left: 20px;  
198         z-index: 1000;  
199     }  
200     langflow-chat {  
201         width: 300px;  
202         height: 440px;  
203     }  
204  
205     </style>  
206 </head>
```

# LangFlow

## API

</> API access

Tweaks (0)



Python    JavaScript    cURL

```
1 import requests
2 url = "https://profanielvieira95-langflow-teste.hf.space/api/v1/run/c74e6ecb-5aea-4bc3-9634-4c3844b1c813" # The complete API endpoint URL for this flow
3
4 # Request payload configuration
5 payload = {
6     "input_value": "hello world!", # The input value to be processed by the flow
7     "output_type": "chat", # Specifies the expected output format
8     "input_type": "chat" # Specifies the input format
9 }
10
11 # Request headers
12 headers = {
13     "Content-Type": "application/json"
14 }
15
16 try:
17     # Send API request
18     response = requests.request("POST", url, json=payload, headers=headers)
```

# LangFlow

## API

```
import requests
```

```
url = "https://profdbielvieira95-langflow-teste.hf.space/api/v1/run/c74e6ecb-5aea-4bc3-9634-4c3844b1c813" #
```

```
The complete API endpoint URL for this flow
```

```
# Request payload configuration
```

```
payload = {
```

```
    "input_value": "hello world!", # The input value to be processed by the flow
```

```
    "output_type": "chat", # Specifies the expected output format
```

```
    "input_type": "chat" # Specifies the input format
```

```
}
```

# LangFlow

```
# Request headers
headers = {
    "Content-Type": "application/json"
}

try:
    # Send API request
    response = requests.request("POST", url, json=payload, headers=headers)
    response.raise_for_status() # Raise exception for bad status codes

    # Print response
    print(response.text)

except requests.exceptions.RequestException as e:
    print(f"Error making API request: {e}")
except ValueError as e:
    print(f"Error parsing response: {e}")
```

# LangFlow

</> API access

Tweaks (0)



Python

JavaScript

cURL

```
1 const payload = {
2     "input_value": "hello world!",
3     "output_type": "chat",
4     "input_type": "chat",
5     // Optional: Use session tracking if needed
6     "session_id": "user_1"
7 };
8
9 const options = {
10     method: 'POST',
11     headers: {
12         'Content-Type': 'application/json'
13     },
14     body: JSON.stringify(payload)
15 };
16
17 fetch('https://profanielvieira95-langflow-teste.hf.space/api/v1/run/c74e6ecb-5aea-4bc3-9634-4c3844b1c813', options)
18     .then(response => response.json())
```

# LangFlow

</> API access

Tweaks (0)



Python    JS    JavaScript    CURL

```
4     "input_type": "chat",
5     // Optional: Use session tracking if needed
6     "session_id": "user_1"
7 };
8
9 const options = {
10   method: 'POST',
11   headers: {
12     'Content-Type': 'application/json'
13   },
14   body: JSON.stringify(payload)
15 };
16
17 fetch('https://profanielvieira95-langflow-teste.hf.space/api/v1/run/c74e6ecb-5aea-4bc3-9634-4c3844b1c813', options)
18   .then(response => response.json())
19   .then(response => console.log(response))
20   .catch(err => console.error(err));
21
```

# LangFlow

</> API access

Tweaks (0)



Python    JS    JavaScript    CURL

```
4     "input_type": "chat",
5     // Optional: Use session tracking if needed
6     "session_id": "user_1"
7 };
8
9 const options = {
10   method: 'POST',
11   headers: {
12     'Content-Type': 'application/json'
13   },
14   body: JSON.stringify(payload)
15 };
16
17 fetch('https://profanielvieira95-langflow-teste.hf.space/api/v1/run/c74e6ecb-5aea-4bc3-9634-4c3844b1c813', options)
18   .then(response => response.json())
19   .then(response => console.log(response))
20   .catch(err => console.error(err));
21
```

# LangFlow - API

## EndPoint da API

<https://profdbielvieira95-langflow-teste.hf.space/api/v1/run/c74e6ecb-5aea-4bc3-9634-4c3844b1c813>

# LangFlow - API

## Teste com o insomnia

The screenshot shows the Insomnia REST Client interface. The top bar displays the application name "Insomnia" and the current session "api.serenatto.cafe". The main window shows a successful POST request with the following details:

- Method: POST
- URL: https://prof.danielvieira95-langflow-t... (partially visible)
- Status: 200 OK
- Time: 2.32 s
- Size: 2.7 KB
- Date: Just Now

The "Body" tab shows the JSON input sent to the API:

```
1 + {  
2   "input_value": "Quais são as variedades de  
3     café disponíveis?",  
4   "output_type": "chat",  
5   "input_type": "chat"  
6 }
```

The "Preview" tab shows the JSON response received from the API:

```
1 + {  
2   "session_id": "c74e6ecb-5aea-4bc3-9634-4c3844b1c813",  
3   "outputs": [  
4     {  
5       "inputs": {  
6         "input_value": "Quais são as variedades de café disponíveis?"  
7       },  
8       "outputs": [  
9         {  
10          "results": {  
11            "message": {  
12              "text_key": "text",  
13              "data": {  
14                "timestamp": "2025-04-28T19:58:27+00:00",  
15                "sender": "Machine",  
16                "sender_name": "AI",  
17                "session_id": "c74e6ecb-5aea-4bc3-9634-4c3844b1c813",  
18                "text": "Trabalhamos com as seguintes variedades de café  
em grãos: Bourbon vermelho, Bourbon amarelo, Blend especial (Mistura de  
Bourbon amarelo e vermelho), Catuai amarelo, Geisha e Virgacheffe.",  
19                "files": [],  
20                "error": false,  
21                "edit": false,  
22                "properties": {  
23                  "text_color": "",  
24                  "background_color": "",  
25                  "edited": false,  
26                  "source": {  
27                    "id": "GrooModel-0th6C",  
28                    "display_name": "Groo",  
29                    "source": "llama3-70b-8192"  
30                  }  
31                }  
32              }  
33            }  
34          }  
35        }  
36      }  
37    }  
38  }  
39 }
```

The bottom status bar indicates "store.books[\*].author".

# Obrigado!

Prof. Me Daniel Vieira

Email: [danielvieira2006@gmail.com](mailto:danielvieira2006@gmail.com)

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

