



ChatBot
Aula 05

Prof. Me Daniel Vieira



Atividade



Atividade

1- Desenvolver um ChatBot especialista em química e física com a Llama Index e Gradio para gerar a interface gráfica da aplicação

Criar a chave da API no Groq e criar o código conforme as aulas anteriores

Código ChatBot

Assistente especialista em Química e Física

Digite sua pergunta sobre Química e Física e receba respostas de IA especializadas!

user_prompt

Defina a primeira Lei de Ohm ?

Clear

output

Uma pergunta clássica!

A primeira Lei de Ohm, também conhecida como Lei de Ohm, é uma lei fundamental da física que relaciona a tensão elétrica (ou diferença de potencial) e a corrente elétrica em um condutor. Foi formulada pelo físico alemão Georg Ohm em 1827.

A Lei de Ohm afirma que, em um condutor isotérmico e homogêneo, a tensão elétrica (V) entre dois pontos é diretamente proporcional à corrente elétrica (I) que flui entre esses pontos, e inversamente proporcional à resistência elétrica (R) do condutor.

Matematicamente, a Lei de Ohm é expressa pela seguinte equação:

$$V = I \times R$$

Onde:

* V é a tensão elétrica (em volts, V)

* I é a corrente elétrica (em ampères, A)

* R é a resistência elétrica (em ohms, Ω)

Atividade

As perguntas para o ChatBot são:

- 1) Quais os componentes do ar ?
- 2) Quais são os produtos químicos da queima de combustível dos automóveis ?
- 3) O que é movimento retilíneo uniforme ?
- 4) Defina a primeira Lei de Ohm ?
- 5) O que são hidrocarbonetos ?
- 6) O que são compostos orgânicos ?
- 7) De que maneira o nitrato e o nitrito ajudam a preparar o solo para a agricultura ?
- 8) O que é eletrólise ?

Atividade

Criar um relatório documentando o código e com print da interface criada e as respostas de cada pergunta e subir o código no GitHub e o relatório no TEAMS

Instalação das bibliotecas

```
pip install gradio
```

```
pip install Groq
```

Exemplo Gradio

- ♦ **Cria a interface gráfica** com `gr.Interface()`, que define como a função `hello()` será exibida e interagida na interface.

Parâmetros:

- `fn=hello`: Define que a função `hello()` será usada para processar os dados.
- `inputs="text"`: Define um campo de entrada **de texto** para o usuário digitar um nome.
- `outputs="text"`: Define que a saída será um texto exibido na tela.

Exemplo Gradio

- `iface.launch()`

Inicia o servidor web e exibe um link

(<http://127.0.0.1:7860/>) onde a interface pode ser acessada no navegador.

Recursos

groqcloud

Playground

Documentation

Metrics

API Keys

Settings

Status

Discord

Chat with us



Personal

Playground

Chat

Studio

llama-3.3-70b-versatile



</> View code

SYSTEM Enter system message (Optional)

Welcome to the Playground

- You can start by typing a message
- Click submit to get a response
- Use the <> icon to view the code

USER

User Message...

Submit



PARAMETERS



Temperature

1

Max Completion Tokens

1024

Stream



JSON Mode





Advanced





Recursos


Clicar em Api Key, atribuir um nome para API e copiar a chave





 Playground


 Documentation


 Metrics


 API Keys

 Settings

 Status





 Discord

 Chat with us

 Personal

API Keys

Manage your API keys. Remember to keep your API keys safe to prevent unauthorized access.

NAME	SECRET KEY	CREATED	LAST USED	USAGE (24HRS)	
GROQ_API	gsk_...vXRm	05/02/2025	Never	0 API Calls	 
groq_api_key	gsk_...B2Zk	05/02/2025	11/02/2025	0 API Calls	 

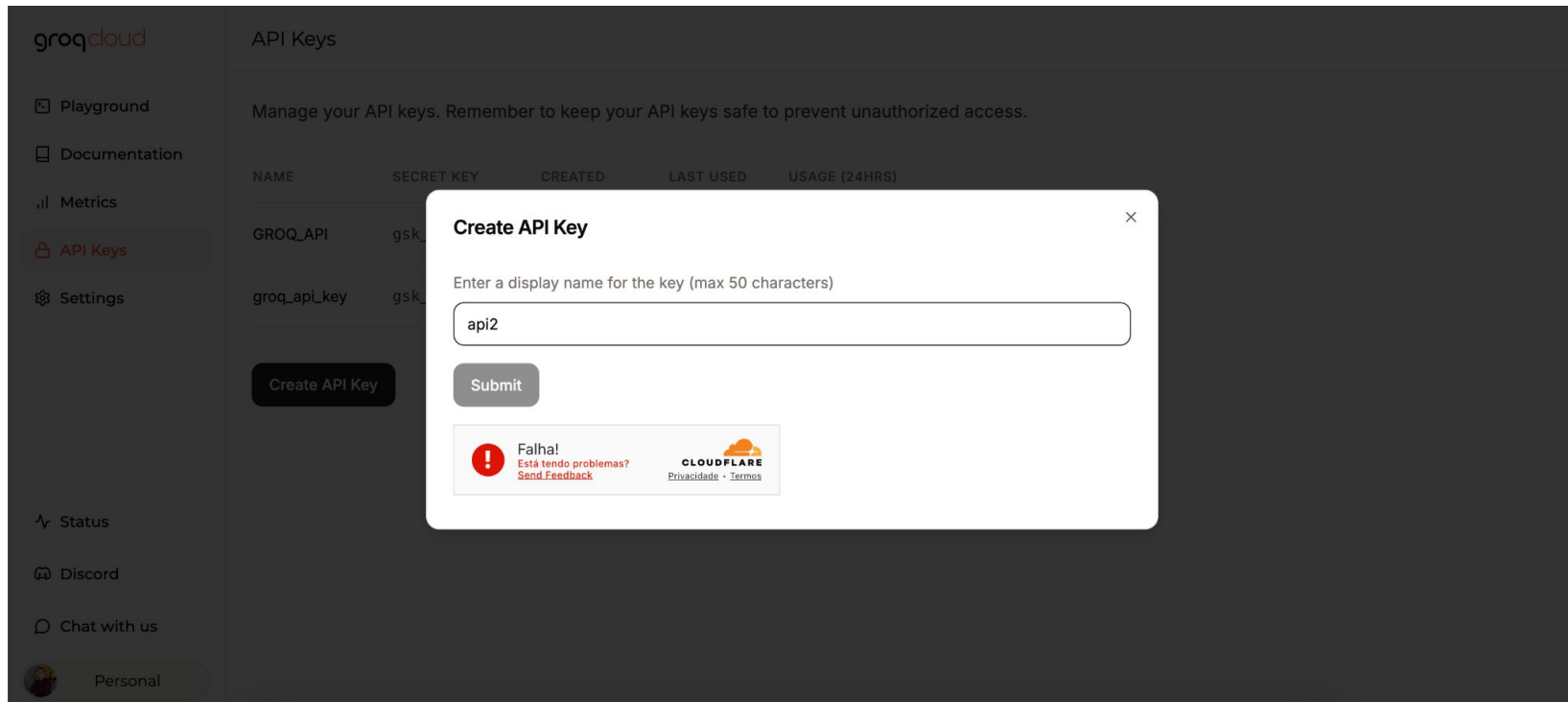
Create API Key

Recursos

Clicar em Api Key, atribuir um nome para API e copiar a chave

Recursos

Clicar em Api Key, atribuir um nome para API clicar em submit



The screenshot shows the Groq Cloud dashboard. On the left is a sidebar with navigation links: Playground, Documentation, Metrics, API Keys (highlighted), Settings, Status, Discord, Chat with us, and a Personal profile section. The main area is titled 'API Keys' and contains a table with columns: NAME, SECRET KEY, CREATED, LAST USED, and USAGE (24HRS). Two API keys are listed: 'GROQ_API' and 'groq_api_key'. A 'Create API Key' button is visible below the table. A modal dialog is open in the center, titled 'Create API Key', with a close button (X) in the top right. Inside the modal, there is a text input field containing 'api2' and a 'Submit' button. Below the input field, there is a red warning icon and the text 'Falha! Está tendo problemas? Send Feedback'. At the bottom of the modal, the Cloudflare logo and links for 'Privacidade' and 'Termos' are displayed.

groqcloud

API Keys

Manage your API keys. Remember to keep your API keys safe to prevent unauthorized access.

NAME	SECRET KEY	CREATED	LAST USED	USAGE (24HRS)
GROQ_API	gsk_...			
groq_api_key	gsk_...			

Create API Key

Create API Key

api2

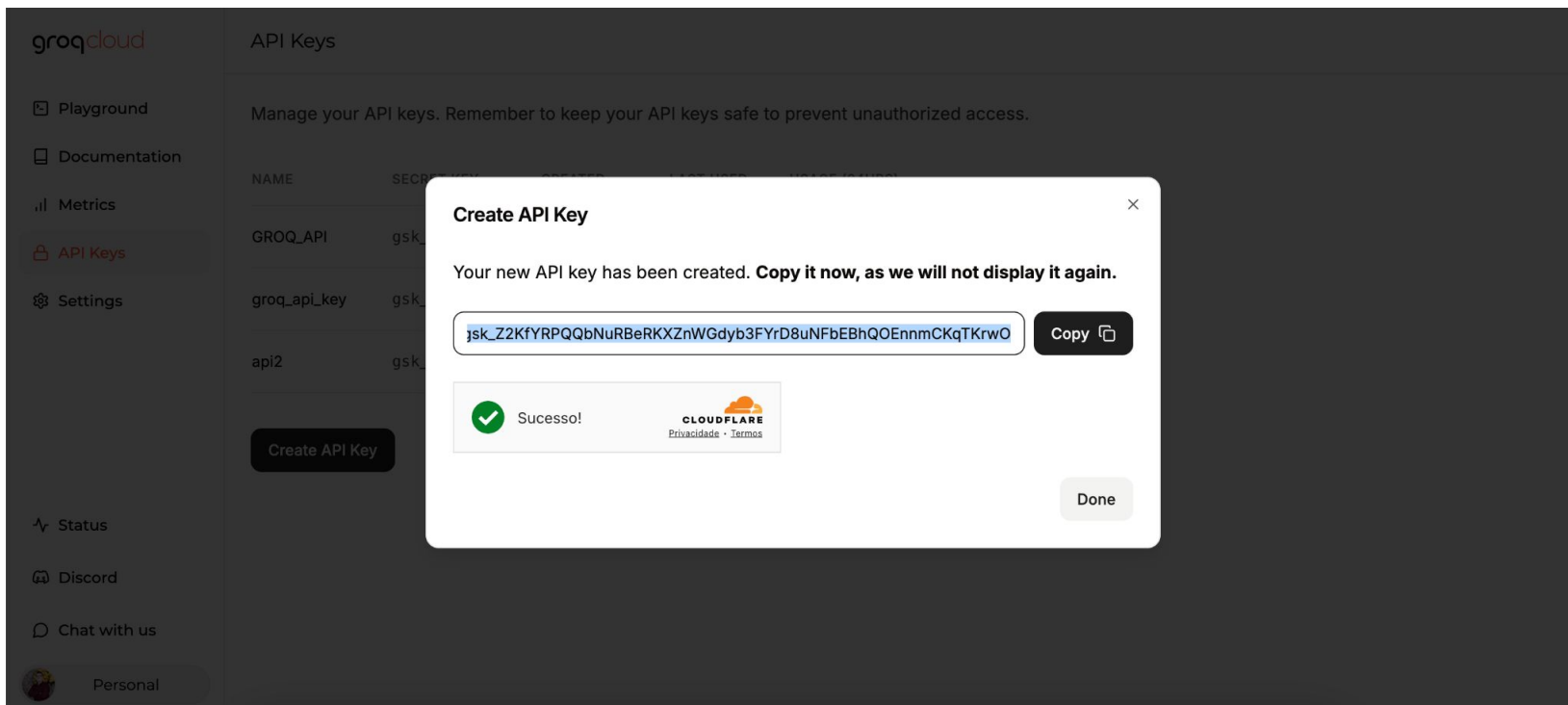
Submit

Falha!
Está tendo problemas?
[Send Feedback](#)

CLOUDFLARE
[Privacidade](#) · [Termos](#)

Recursos

Copiar a API gerada e salvar, pois precisaremos dela



The screenshot shows the Groq Cloud interface. On the left is a sidebar with navigation links: Playground, Documentation, Metrics, API Keys (highlighted), Settings, Status, Discord, Chat with us, and a Personal profile section. The main area is titled 'API Keys' and contains a table with columns 'NAME' and 'SECRET'. The table lists three keys: 'GROQ_API' (secret: gsk...), 'groq_api_key' (secret: gsk...), and 'api2' (secret: gsk...). A 'Create API Key' button is at the bottom left of the table. A modal window titled 'Create API Key' is open in the center. It contains the text: 'Your new API key has been created. Copy it now, as we will not display it again.' Below this is a text input field containing the key 'jsk_Z2KfYRPQQbNuRBeRKXZnWGdyb3FYrD8uNFbEBhQOEnnmCKqTKrwO'. To the right of the input is a 'Copy' button with a clipboard icon. At the bottom of the modal is a 'Done' button. A success message at the bottom left of the modal reads 'Sucesso!' with a green checkmark icon and the Cloudflare logo and 'Privacidade - Termos' link.

groqcloud

API Keys

Manage your API keys. Remember to keep your API keys safe to prevent unauthorized access.

NAME	SECRET
GROQ_API	gsk...
groq_api_key	gsk...
api2	gsk...

Create API Key

Create API Key

Your new API key has been created. **Copy it now, as we will not display it again.**

jsk_Z2KfYRPQQbNuRBeRKXZnWGdyb3FYrD8uNFbEBhQOEnnmCKqTKrwO

Copiar

Sucesso!

CLOUDFLARE
Privacidade - Termos

Done

Código ChatBot

Aula04prep > aula04.ipynb > ...

+ Code + Markdown | ▶ Run All ↺ Restart ≡ Clear All Outputs | 📄 Jupyter Variables ≡ Outline ...

Python 3.13.1

+ Code + Markdown

≡ ▶ ↻ 📄 ... 🗑

```
import os # Biblioteca para interagir com o sistema operacional
import gradio as gr # Importa Gradio para criar a interface web
from groq import Groq # Importa o cliente da API Groq

# Defina sua chave da API do GROQ
GROQ_API_KEY = os.getenv('GROQ_API_KEY', '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') # Substitua pela sua chave

# Inicializa o cliente da API Groq
client = Groq(api_key=GROQ_API_KEY)

# Função para processar a entrada do usuário e gerar uma resposta
def assistente_agricultura(user_prompt):
    if user_prompt.strip() == "15":
        return "Encerrando assistente Agro! Até mais! 🌱"

    # Faz a solicitação ao modelo
    completion = client.chat.completions.create(
        model="llama3-8b-8192",
        messages=[
            {"role": "system", "content": "Você é um assistente especializado em agricultura."},
            {"role": "user", "content": user_prompt}
        ],
        temperature=0,
        max_tokens=1024,
        top_p=1,
        stream=False
    )

    # Retorna a resposta do assistente
    return completion.choices[0].message.content
```

Código ChatBot

```
# Configuração da interface Gradio
iface = gr.Interface(
    fn=assistente_agricultura,
    inputs=gr.Textbox(lines=2, placeholder="Digite sua pergunta sobre agricultura..."),
    outputs="text",
    title="🌾 Assistente Agro IA",
    description="Digite sua pergunta sobre agricultura e receba respostas de IA especializadas! 🌾",
    live=True
)

# Executa a interface web
if __name__ == "__main__":
    iface.launch()
```

[1] ✓ 1.6s

Python

... [/Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages/tqdm/auto.py:21](#): TqdmWarning: IPProgress not found. Please update jupyter and ipywidgets. See <https://ipywidgets.readthedocs.io/en/latest/faq.html#ipynb-notebook-cannot-find-ipywidgets>

from .autonotebook import tqdm as notebook_tqdm

* Running on local URL: <http://127.0.0.1:7860>

To create a public link, set `share=True` in `launch()`.

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

