



ChatBot
Aula 07

Prof. Me Daniel Vieira

SENAI

Atividade

1- Criar um chatbot para realizar a leitura de um pdf com informações sobre agricultura, tipos de solo, que auxilie o agricultor na escolha da semente e como preparar o solo para o plantio, bem como definir os limites de temperatura para o solo e irrigação de acordo com a semente plantada.

Atividade

Você deve criar um banco de dados vetorial para armazenar as informações do pdf carregado conforme o exemplo da aula anterior

Perguntas a serem feitas para chatbot assistente Agro

- 1) Por que é importante preparar o solo adequadamente antes do plantio?
- 2) Quais são os principais tipos de solo e como prepará-los para o cultivo?
- 3) Quais sensores podem ser utilizados para monitorar a umidade do solo e como funcionam?
- 4) Como é possível ajustar o pH do solo para atender às necessidades das diferentes culturas?
- 5) Quais são os principais fertilizantes utilizados e qual a diferença entre fertilizantes orgânicos e químicos?
- 6) Qual é o processo recomendado para a correção e preparo do solo antes do plantio?
- 7) Como garantir a fertilidade do solo para diferentes tipos de culturas?
- 8) Quais são os principais fatores que devem ser levados em consideração ao planejar a irrigação do solo?
- 9) Quais são as melhores práticas para evitar a compactação do solo e a erosão?
- 10) Onde um agricultor pode obter mais informações sobre preparo do solo e escolha de sementes?

Atividade

Subir o código no Teams e criar um relatório documentando os passos para criar o chatbot

ChatBot com RAG



Retrieval - Augmented Generation

Tecnicamente RAG é uma estrutura de IA para recuperar fatos de uma base de conhecimento externa para fundamentar grandes modelos de linguagem (LLMs) nas informações mais precisas e atualizadas e para fornecer aos usuários uma visão sobre o processo gerador de LLMs.

RAG

Os LLMs enfrentam uma série de desafios atualmente, como os seguintes:

- Fornecer informações falsas quando não possuem uma resposta adequada.
- Oferecer informações desatualizadas ou genéricas quando os usuários solicitam respostas específicas e atualizadas.
- Gerar respostas baseadas em fontes não confiáveis.
- Criar respostas imprecisas devido à confusão terminológica, quando diferentes fontes de treinamento utilizam a mesma terminologia para descrever conceitos distintos.

Chroma DB

ChromaDB é um banco de dados vetorial de código aberto projetado para armazenar, indexar e recuperar vetores de alta dimensão de maneira eficiente. Ele é amplamente utilizado para tarefas como **busca semântica, sistemas de recomendação e aplicações de IA generativa**, onde embeddings de texto, imagem ou áudio são usados para recuperar informações relevantes.

Chroma DB

Principais Características do ChromaDB

- 1. Armazenamento de Embeddings** - Ideal para trabalhar com modelos de aprendizado de máquina e IA, como OpenAI, LlamaIndex e LangChain.
- 2. Pesquisa Vetorial Rápida** - Suporte a algoritmos eficientes de busca aproximada (ANN - Approximate Nearest Neighbors).
- 3. Persistência de Dados** - Pode armazenar embeddings de forma persistente para não precisar recalcular a cada consulta.
- 4. Interface Simples via Python** - Fácil integração em projetos de IA, principalmente em aplicações que utilizam embeddings de texto.
- 5. Integração com Frameworks de IA** - Funciona bem com ferramentas como LangChain, LlamaIndex e FAISS.

Código

```
1 #instalando a biblioteca Llama index
2 !pip install -q llama-index

[3] 1 # biblioteca para permitir a leitura de arquivos
2 from llama_index.core import SimpleDirectoryReader

[4] 1 documentos = SimpleDirectoryReader(input_dir='documentos')

[5] 1 documentos.input_files
→ [PosixPath('/content/documentos/serenatto_cafes_especiais.pdf')]

[6] 1 docs = documentos.load_data()

[7] 1 docs
→ \nágua\n \nquente\n \npara\n \nremover\n \nqualquer\n \nsabor\n \nde\n \npapel\n \ne\n \npré-aquecer\n \no\n \nsuporte.\n 5. Adição do café: Despeje o pó no filtro.
6. Infusão: Adicione a água quente lentamente sobre o café, preferencialmente em \nciclos,\n \ncomecando\n \ndo\n \ncentro\n \ne\n \nmovendo\n \npara\n \nfora\n \nem\n \nmovimento\n \nespiral.\n \nIsso\n \njuda\n \na\n \ngarantir\n \numa\n \nextração\n \nuniforme\n \nde\n \ntodos\n \nos\n \nsabores\n \ne\n \naromas.\n 7.
Finalização: Após a passagem da água, remova o filtro e descarte-o. Agora é só \nservir\n \ne\n \ndesfrutar\n \ndo\n \nseu\n \ndelicioso\n \ncafé.\n P: Que tipo de água devo usar para fazer café? R: Recomendamos usar água filtrada com baixo teor de minerais. A qualidade da água \ntem\n \num\n \ngrande\n \nimpacto\n \nono\n \nsabor\n \nfinal\n \ndo\n \ncafé,\n \npor\n \nissos\n \nágua\n \npura,\n \nsem\n \ncloro\n \nou\n \nimpurezas,\n \nné\n \nessencial.\n      , path=None,
url=None, mimetype=None), image_resource=None, audio_resource=None, video_resource=None, text_template='{metadata_str}\n\n{content}'),
Document(id = '5b954e8d-d50f-42b0-a702-395779a15067', embedding=None, metadata={'page_label': '8', 'file name': 'serenatto_cafes especiais.pdf', 'file path':
Em execução (21m42s) <cell line: 0> launch() > block_thread()
```

Código

Verificando informações do documento carregado

```
[8] 1 len(docs)
→ 10

[9] 1 print(docs[0].get_metadata_str())
→ page_label: 1
file_name: serenatto_cafes_especiais.pdf
file_path: /content/documentos/serenatto_cafes_especiais.pdf
file_type: application/pdf
file_size: 133957
creation_date: 2025-03-12
last_modified_date: 2025-03-12

[10] 1 docs[0].__dict__
→ {'id_': '35786cd1-b5ff-43f0-9696-192685e9c533',
'embedding': None,
'metadata': {'page_label': '1',
'file_name': 'serenatto_cafes_especiais.pdf',
'file_path': '/content/documentos/serenatto_cafes_especiais.pdf',
'file_type': 'application/pdf',
'file_size': 133957,
'creation_date': '2025-03-12',
'last_modified_date': '2025-03-12'},
'excluded_embed_metadata_keys': ['file_name',
'file_type',
'file_size',
'creation date']}
```

Código

importando bibliotecas

[11] O código gerado pode estar sujeito a uma licença | Boxiang7JC/llama_index_boxiang

```
1 # importando a biblioteca
2 from llama_index.core.node_parser import SentenceSplitter
3 node_parser= SentenceSplitter(chunk_size=1200)
```

```
[12] 1 nodes = node_parser.get_nodes_from_documents(docs, show_progress=True)
```

➡ Parsing nodes: 100%

[13] 1 nodes

```
→ TextNode(id_='fca8b795-4f20-4935-87e1-83f470eca909', embedding=None, metadata={'page_label': '8', 'file_name': 'serenatto_cafes_especiais.pdf', 'file_path': '/content/documentos/serenatto_cafes_especiais.pdf', 'file_type': 'application/pdf', 'file_size': 133957, 'creation_date': '2025-03-12', 'last_modified_date': '2025-03-12'}, excluded_embed_metadata_keys=['file_name', 'file_type', 'file_size', 'creation_date', 'last_modified_date', 'last_accessed_date'], excluded_llm_metadata_keys=['file_name', 'file_type', 'file_size', 'creation_date', 'last_modified_date', 'last_accessed_date'], relationships=[<NodeRelationship.SOURCE: '1'>: RelatedNodeInfo(node_id='5b954e8d-d50f-42b0-a702-395779a15067', node_type=<ObjectType.DOCUMENT: '4'>, metadata={'page_label': '8', 'file_name': 'serenatto_cafes_especiais.pdf', 'file_path': '/content/documentos/serenatto_cafes_especiais.pdf', 'file_type': 'application/pdf', 'file_size': 133957, 'creation_date': '2025-03-12', 'last_modified_date': '2025-03-12'}, hash='281a43472b035de033ca5d67c661522c00898eb0ec27cd7722cf6814cf12cd65'), metadata_template='{key}: {value}', metadata_separator='\n', text='Café gelado P: Como posso preparar café gelado? R: Para preparar um delicioso café gelado, você pode usar o método do coado, \ncontendo\n\n40%\n\nndo\n\nvolume\n\nem\n\nngelo\n\nne\n\n60%\n\nndo\n\nvolume\n\nem\n\nágua\n\ninque\n\nseguindo\n\nnas\n\nseguintes\n\netapas:\n\n1. Pesagem do café: Use a proporção de 1 grama de café para cada 16 mL de água\nntotal\n2. Moagem do Café: Moa o café levemente mais fino do que
```

Em execução (24m1s) <cell line: 0> launch() > block_thread()

Código

```
[14] 1 len(nodes)
```

```
→ 10
```

```
[15] 1 nodes [0]
```

```
→ TextNode(id_='757a9a46-5ba3-4356-97f9-b192fcce7a37', embedding=None, metadata={'page_label': '1', 'file_name': 'serenatto_cafes_especiais.pdf', 'file_path': '/content/documentos/serenatto_cafes_especiais.pdf', 'file_type': 'application/pdf', 'file_size': 133957, 'creation_date': '2025-03-12', 'last_modified_date': '2025-03-12'}, excluded_embed_metadata_keys=['file_name', 'file_type', 'file_size', 'creation_date', 'last_modified_date', 'last_accessed_date'], excluded_llm_metadata_keys=['file_name', 'file_type', 'file_size', 'creation_date', 'last_modified_date', 'last_accessed_date'], relationships={<NodeRelationship.SOURCE: '1': RelatedNodeInfo(node_id='35786cd1-b5ff-43f0-9696-192685e9c533', node_type=<ObjectType.DOCUMENT: '4'), metadata={'page_label': '1', 'file_name': 'serenatto_cafes_especiais.pdf', 'file_path': '/content/documentos/serenatto_cafes_especiais.pdf', 'file_type': 'application/pdf', 'file_size': 133957, 'creation_date': '2025-03-12', 'last_modified_date': '2025-03-12'}, hash='820b277edf233136c5af69d8b63f74658d9de4c25893c4e5d205bcb010e8b2e'}}, metadata_template='{key}: {value}', metadata_separator='\n', text='Serenatto – Loja online de cafés especiais Informações gerais P: Quais produtos são vendidos no site da Serenatto? R: A Serenatto é uma loja online especializada na venda de grãos de café especiais, \noferecendo\n\ngrãos\n\ntorrados,\n\nprovenientes\n\nde\nnfazendas\n\nselecionadas\n\nde\nMinas\n\nGerais.\n\nTodos\n\nos\n\nossos\nncafés\nn\npossuem\nn\nnota\nn\nSCA\nn\nacima\nnde\nn\n80,\n\ngarantindo\nn\nalta\nn\nqualidade\nn\nperfis\nn\nsensoriais\nn\nexcepcionais.\n\n P: De onde vêm os grãos de café da Serenatto? R: Nossos grãos são cuidadosamente selecionados de fazendas localizadas em Minas Gerais,\n\nconhecidas\n\npel\n\nalta\nn\nqualidade\nn\nnde\nn\nseus\nn\ncafés\nn\nespeciais.\n\n nCada\nn\nfazenda\nn\né\nn\nescolhida\nn\npor\nn\nsuas\nn\npráticas\nn\nsustentáveis\nn\npel\n\nncuidado\nn\nexcepcional\nn\ncom\nn\na\nn\nprodução.\n\n P: Qual é a quantidade vendida? R: Os grãos já torrados são vendidos em pacotes de 250g P: Quais são as variedades de café disponíveis? R: Trabalhamos com as seguintes variedades de café em grãos: ● Bourbon vermelho ● Bourbon amarelo ● Blend especial (Mistura de Bourbon amarelo e vermelho) ● Catuá amarelo ● Geisha ● Yirgacheffe P: Quais são os preços de cada variedade de café? R: Os preços são os seguintes: ● Bourbon vermelho: R$ 41,00 ● Bourbon amarelo: R$ 43,00 ● Blend especial: R$ 37,50 ● Catuá amarelo: R$ 55,00 ● Geisha: R$ 105,00 ● Yirgacheffe: R$ 110,00', mimetype='text/plain', start_char_idx=0, end_char_idx=1515, metadata_seperator='\n', text_template='{metadata_str}\n\n{content}')
```

```
[16] 1 nodes [9]
```

```
→ TextNode(id_='22844b30-fd49-4dc8-a6b3-8d0f69578f1a', embedding=None, metadata={'page_label': '10', 'file_name': 'serenatto_cafes_especiais.pdf', 'file_path':
```

Código

[16] 1 nodes [9]

```
→ TextNode(id_='22844b30-fd49-4dc8-a6b3-8d0f69578f1a', embedding=None, metadata={'page_label': '10', 'file_name': 'serenatto_cafes_especiais.pdf', 'file_path': '/content/documentos/serenatto_cafes_especiais.pdf', 'file_type': 'application/pdf', 'file_size': 133957, 'creation_date': '2025-03-12', 'last_modified_date': '2025-03-12'}, excluded_embed_metadata_keys=['file_name', 'file_type', 'file_size', 'creation_date', 'last_modified_date', 'last_accessed_date'], excluded_llm_metadata_keys=['file_name', 'file_type', 'file_size', 'creation_date', 'last_modified_date', 'last_accessed_date'], relationships={<NodeRelationship.SOURCE: '1': RelatedNodeInfo(node_id='b079151c-191c-47a3-8c7d-8e7ac52011c8', node_type=<ObjectType.DOCUMENT: '4'>, metadata={'page_label': '10', 'file_name': 'serenatto_cafes_especiais.pdf', 'file_path': '/content/documentos/serenatto_cafes_especiais.pdf', 'file_type': 'application/pdf', 'file_size': 133957, 'creation_date': '2025-03-12', 'last_modified_date': '2025-03-12'}, hash='54ca4fc7b770a7c832c0b03e122cb6d0dfd82241edbe306a0161b80d8ed17a5a')}, metadata_template='{key}: {value}', metadata_separator='\n', text='Armazenamento e validade P:  
Como devo armazenar os grãos de café para manter a qualidade? R: Recomendamos armazenar os grãos em um recipiente hermético, longe da umidade \ne\n \ncalor.\n \nEvite\n \nguardar\n \no\n \ncafér\n \no\n \nnaf\n \nncaldeira,\n \npois\n \na\n \numidade\n \ne\n \nas\n \nvariações\n \nde\n \ntemperatura\n \npodem\n \nprejudicar\n \no\n \nsabor\n \ne\n \no\n \naroma.\n P: Qual é a validade dos cafés da Serenatto? R: Nossos cafés são enviados recém-torrados para garantir frescor. Para uma \nexperiência\n \nmelhor\n \ncom\n \nas\n \ncaracterísticas\n \ndas\n \nnotas\n \nsensoriais\n \ne\n \na\n \nintensidade\n \ndo\n \naroma,\n \no\n \nconsumo\n \ndeve\n \nser\n \nenentre\n \ntrês\n \ne\n \nquatro\n \nmeses.\n \nNo\n \nnentanto,\n \ndepois\n \nde\n \nnaberto,\n \no\n \nconsumo\n \ndeve\n \nser,\n \nde\n \npreferência,\n \nenentre\n \ndas\n \n(quando\n \nestiver\n \nmoído)\n \na\n \nnquatro\n \nsemanas\n \n(em\n \ngrãos)\n \nda\n \nndata\n \nda\n \ntorra.', mimetype='text/plain', start_char_idx=0, end_char_idx=884, metadata_seperator='\n', text_template='{metadata_str}\n{content}')'
```

Código

```
[17] 1 #instalação do hugging face
2 !pip install -q llama-index-embeddings-huggingface

[18] 1 from llama_index.embeddings.huggingface import HuggingFaceEmbedding

[19] 1 # Classe personalizada para integrar ao Chroma DB
2
3 class ChromaEmbeddingWrapper:
4     def __init__(self, model_name:str): # inicializa com o modelo embedding
5         self.model = HuggingFaceEmbedding(model_name = model_name)
6
7
8     def __call__(self, input): # Converte a entrada para o formato compativel com o HuggingFace
9         return self.model.embed(input)
10
11
12
```

Código

```
[20]   1 embed_model_chroma= ChromaEmbeddingWrapper(model_name='intfloat/multilingual-e5-large')
```

```
→ /usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:  
The secret `HF_TOKEN` does not exist in your Colab secrets.  
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart you  
You will be able to reuse this secret in all of your notebooks.  
Please note that authentication is recommended but still optional to access public models or datasets.  
  warnings.warn(
```

Código

Instalação do Chroma DB

```
✓ [21] 1 #instalação do chroma db
         2 !pip install -q llama-index-vector-stores-chroma

✓ [22] 1 #importando chroma db
         2
         3 import chromadb
         4 db = chromadb.PersistentClient(path='./chroma_db')
         5 chroma_client = db
         6 collection_name = 'documentos_serenatto'
         7
         8 try:
         9     chroma_collection = chroma_client.get_or_create_collection(
          10         name=collection_name,
          11         embedding_function=embed_model_chroma)
          12
          13 except Exception as e :
          14     print(f'Erro ao criar coleção {e}')
          15
```

Código

```
✓ 0s [23] 1 from llama_index.vector_stores.chroma import ChromaVectorStore  
2 from llama_index.core import StorageContext  
  
✓ 0s [24] 1 #criando a coleção do banco de dados  
2 vector_store = ChromaVectorStore(chroma_collection=chroma_collection)  
3 storage_context = StorageContext.from_defaults(vector_store=vector_store)  
  
✓ 5s [25] 1 embed_model = HuggingFaceEmbedding(model_name='intfloat/multilingual-e5-large')  
  
✓ 0s [26] 1 from llama_index.core import VectorStoreIndex  
  
✓ min [27] 1 index = VectorStoreIndex(nodes, storage_context=storage_context, embed_model = embed_model)
```

Código

A api do groq deve ser copiada da plataforma Groq conforme as aulas anteriores

```
✓ [28] 1 from llama_index.core import load_index_from_storage
        2 index = load_index_from_storage(storage_context, embed_model = embed_model)

✓ [29] 1 from google.colab import userdata
        2 Groq_api = userdata.get('api_groq_chat')

✓ [30] 1 !pip install -q llama-index-llms-groq

✓ [32] 1 from llama_index.llms.groq import Groq

✓ [33] 1 llms = Groq(model = 'llama3-70b-8192',api_key=Groq_api )

✓ [34] 1 query_engine = index.as_query_engine(llm=llms,similarity_top_k=2)

✓ [35] 1 query_engine.query('Quais grãos estão disponíveis').response
→ 'Trabalhamos com as seguintes variedades de café em grãos: Bourbon vermelho, Bourbon amarelo, Blend especial (Mistura de Bourbon amarelo e vermelho), Catuaí amarelo, Geisha, Yirgacheffe.'

✓ [36] 1 query_embedding = embed_model.get_text_embedding('Quais grãos estão disponíveis ?')
```

Código

```
✓ [37]   1 chroma_collection.query(query_embedding,n_results=2,include=['distances','embeddings'])
```

```
→ {'ids':[['3591a1e4-813c-45a4-b0eb-13ca928fb822',
    '757a9a46-5ba3-4356-97f9-b192fcce7a37']],
    'embeddings': [array([[ 0.01460024, -0.0119941, -0.00576889, ...,  0.00555894,
        -0.0503395 ,  0.00342874],
       [ 0.01460024, -0.0119941, -0.00576889, ...,  0.00555894,
        -0.0503395 ,  0.00342874]]),
    'documents': None,
    'uris': None,
    'data': None,
    'metadatas': None,
    'distances': [[0.3871879346745659, 0.3871879346745659]],
    'included': [<IncludeEnum.embeddings: 'embeddings'>,
    <IncludeEnum.distances: 'distances'>]}
```

```
✓ [38]   1 chat_engine = index.as_chat_engine(mode='context',llm = llms)
```

```
✓ [39]   1 pergunta = chat_engine.chat('Quais grãos estão disponíveis ?').response
2 print(pergunta)
```

```
→ Os grãos disponíveis são Bourbon vermelho, Bourbon amarelo, Blend especial (uma mistura de Bourbon amarelo e vermelho), Catuai amarelo, Geisha e Yirgacheffe.
```

```
[ ]   1 Comece a programar ou gere código com IA.
```

Query Engine e Chat Engine

Característica	Query Engine	Chat Engine
Propósito	Recuperação de informações específicas	Interação conversacional contextualizada
Contexto	Não mantém contexto de conversa	Mantém contexto de conversa
Respostas	Diretas e baseadas em dados	Dinâmicas e contextualizadas
Uso Ideal	Perguntas pontuais e específicas	Conversas contínuas e interativas

Código

→ 'Os grãos disponíveis são: Bourbon vermelho, Bourbon amarelo, Blend especial (Mistura de Bourbon amarelo e vermelho), Catuaí amarelo, Geisha e Yirgacheffe.'

✓ [40] 1 pergunta = chat_engine.chat('Quais sao os preços dos graos ?').response
2 print(pergunta)

→ Os preços dos grãos são: Bourbon vermelho: R\$ 41,00, Bourbon amarelo: R\$ 43,00, Blend especial: R\$ 37,50, Catuaí amarelo: R\$ 55,00, Geisha: R\$ 105,00, e Yirgacheffe: R\$ 110,00

[] 1 Comece a programar ou gere código com IA.

✓ [41] 1 pergunta = chat_engine.chat('Voce pode me dar mais detalhes sobre o Catuai amarelo ?').response
2 print(pergunta)

→ O Catuai amarelo tem doçura máxima (5/5) e corpo médio-alto (4/5) com amargor baixo (1/5). Além disso, apresenta uma acidez expressiva e notas incríveis de pitanga, desviando

[] 1 Comece a programar ou gere código com IA.

✓ [42] 1 pergunta = chat_engine.chat('Qual o preço dele ?').response
2 print(pergunta)

→ O preço do Catuai amarelo é R\$ 55,00.

Código

Verificando a memória do Chat

[43] 1 chat_engine.chat_history

```
→ [ChatMessage(role=<MessageRole.USER: 'user'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='Quais grãos estão disponíveis ?')]),
 ChatMessage(role=<MessageRole.ASSISTANT: 'assistant'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='Os grãos disponíveis são Bourbon vermelho, Bourbon amarelo, Blend especial (uma mistura de Bourbon amarelo e vermelho), Catuai amarelo, Geisha e Yirgacheffe.')]),
 ChatMessage(role=<MessageRole.USER: 'user'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='Quais são os preços dos grãos ?')]),
 ChatMessage(role=<MessageRole.ASSISTANT: 'assistant'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='Os preços dos grãos são: Bourbon vermelho: R$ 41,00, Bourbon amarelo: R$ 43,00, Blend especial: R$ 37,50, Catuai amarelo: R$ 55,00, Geisha: R$ 105,00, e Yirgacheffe: R$ 110,00.')]),
 ChatMessage(role=<MessageRole.USER: 'user'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='Você pode me dar mais detalhes sobre o Catuai amarelo ?')]),
 ChatMessage(role=<MessageRole.ASSISTANT: 'assistant'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='O Catuai amarelo tem doçura máxima (5/5) e corpo médio-alto (4/5) com amargor baixo (1/5). Além disso, apresenta uma acidez expressiva e notas incríveis de pitanga, desviando do tradicional.')]),
 ChatMessage(role=<MessageRole.USER: 'user'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='Qual o preço dele ?')]),
 ChatMessage(role=<MessageRole.ASSISTANT: 'assistant'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='O preço do Catuai amarelo é R$ 55,00.')])]
```

Código

```
[44] 1 from llama_index.core.memory import ChatSummaryMemoryBuffer
[45] 1 memory = ChatSummaryMemoryBuffer(llm=llms,token_limit=256)
[46] 1 chat_engine= index.as_chat_engine(
[47] 2     chat_mode = 'context',
[48] 3     llm=llms,
[49] 4     memory=memory,
[50] 5     system_prompt=( '''Voce é especialista em cafes da loja Serenato, uma loja online que vende graos de cafe
[51] 6     torrados, sua função é tirar duvidas de forma simpatica e natural sobre os graos disponiveis''' )
[52] 7 )
[53]
[54] [47]: 1 response = chat_engine.chat('Olá')
[55] 2 print(response)
[56]
[57] ➔ Olá! Bem-vindo à loja online Serenatto! Eu sou o especialista em cafés aqui. Estou aqui para ajudar a tirar suas dúvidas sobre nossos deliciosos grãos de café torrados. Qual
[58]
[59] [48]: 1 response = chat_engine.chat('Voce pode me dar mais detalhes sobre o catuai amarelo')
[60] 2 print(response)
[61]
[62] ➔ 0 Catuai amarelo! É um dos nossos cafés mais exóticos e especiais. Ele possui uma docura máxima (5/5) e um corpo médio-alto (4/5), com um amargor baixo (1/5). O que o torna r
[63] Em execução (33m51s) <cell line: 0> > launch() > block_thread()
```

Código

```
[48] 1 response = chat_engine.chat('Voce pode me dar mais detalhes sobre o catui amarelo')
2 print(response)

→ O Catuai amarelo! É um dos nossos cafés mais exóticos e especiais. Ele possui uma doçura máxima (5/5) e um corpo médio-alto (4/5), com um amargor baixo (1/5). O que o torna r...
    O processo de produção do Catuai amarelo é fermentado, cultivado em altitude de 1.200m, com torra média. Isso lhe confere um perfil sensorial muito interessante e agradável.
    Se você é um amante de cafés que oferecem algo novo e emocionante, o Catuai amarelo é uma ótima escolha. Você vai adorar sua complexidade e sua capacidade de surpreender!
```

```
[49] 1 response = chat_engine.chat('qual o preço dele')
2 print(response)

→ O preço do Catuai amarelo é de R$ 55,00 por pacote de 250g. É um investimento vale a pena para quem busca uma experiência de café única e de alta qualidade. Além disso, como...
```

```
[50] 1 memory.get()

→ [ChatMessage(role=<MessageRole.SYSTEM: 'system'>, additional_kwargs={}, blocks=[TextBlock(block_type='text', text='It looks like we have a conversation started in Portuguese!\n\nTo summarize, the user greeted with "Ola" and the assistant responded with a warm welcome to the online store Serenatto, introducing themselves as a coffee expert. The assistant is ready to help answer any questions the user may have about their delicious roasted coffee beans.\n\nThe conversation so far:\n\nUser: Ola\nAssistant: Olá! Bem-vindo à loja online Serenatto! Eu sou o especialista em cafés aqui. Estou aqui para ajudar a tirar suas dúvidas sobre nossos deliciosos grãos de café torrados. Qual...')]
```

Código

```
✓ [51] 1 #resetando o chat  
2 chat_engine.reset()
```

```
✓ [52] 1 response = chat_engine.chat('qual o preço dele')  
2 print(response)
```

Olá!

Você está perguntando sobre o preço de um dos nossos cafés especiais, certo?

Nós temos várias opções disponíveis, e os preços variam de acordo com a variedade. Aqui vão os preços de cada uma delas:

- * Bourbon Vermelho: R\$ 41,00
- * Bourbon Amarelo: R\$ 43,00
- * Blend Especial (mistura de Bourbon Amarelo e Vermelho): R\$ 37,50
- * Catuaí Amarelo: R\$ 55,00
- * Geisha: R\$ 105,00
- * Yirgacheffe: R\$ 110,00

Qual variedade você está interessado em saber mais sobre?

Código

Criando funções para o chatbot

```
[ ]    1 # implementando a interface com o Gradio
2 !pip install gradio

✓ [53]   1 import gradio as gr

✓ [55]   1 # Criando a função para conversar com o chatbot
2 def converse_com_bot(message, chat_history):
3     response = chat_engine.chat(message)
4
5     if chat_history is None:
6         chat_history = []
7
8     chat_history.append({"role": "user", "content": message})
9     chat_history.append({"role": "assistant", "content": response.response})
10
11    return "", chat_history
12
13 # Criando a função para resetar o chatbot
14 def resetar_chat():
15     chat_engine.reset()
16     return []

17
```

Código

Criando funções para o chatbot

✓ [55] 19
0s

```
1 with gr.Blocks() as app:  
2  
3     gr.Markdown('# Chatbot da Serenatto')  
4     chatbot = gr.Chatbot(type='messages')  
5     msg = gr.Textbox(label='Digite a sua mensagem')  
6     limpar = gr.Button('Limpar')  
7  
8     msg.submit(converse_com_bot, [msg,chatbot], [msg,chatbot])  
9     limpar.click(resetar_chat,None,chatbot,queue=False)  
10  
11     app.launch(debug=True)
```

... Running Gradio in a Colab notebook requires sharing enabled. Automatically setting `share=True` (you can turn this off by setting `share=False` in `launch()` explicitly).

Colab notebook detected. This cell will run indefinitely so that you can see errors and logs. To turn off, set debug=False in launch().

* Running on public URL: <https://a001597c67037ec64f.gradio.live>

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spa

Chatbot da Serenatto



Interface ChatBot

Chatbot da Serenatto

The screenshot shows a conversational interface. At the top left, there's a small icon labeled 'Chatbot'. On the right, a message bubble contains the text 'voce pode me recomendar um cafe doce e suave'. Below this, the bot responds with 'Sim!'. It then provides a detailed recommendation about Bourbon Vermelho coffee, mentioning its maximum sweetness (5/5), body (5/5), low acidity (1/5), light bitterness (2/5), and chocolate notes. It also asks if the user would like to try it. At the bottom, there's a text input field with the placeholder 'Digite a sua mensagem'.

Chatbot

voce pode me recomendar um cafe doce e suave

Sim!

Se você procura um café doce e suave, eu recomendaria o Bourbon Vermelho. Ele tem uma doçura máxima (5/5) e um corpo encorpado (5/5), o que o torna uma opção ideal para quem busca um café suave e agradável.

Além disso, o Bourbon Vermelho tem uma acidez baixa (1/5) e um amargor leve (2/5), o que o torna uma opção perfeita para quem não gosta de cafés muito ácidos ou amargos.

Ele também tem notas de chocolate, o que adiciona uma camada de complexidade ao seu perfil sensorial.

Você gostaria de experimentar o Bourbon Vermelho?

voce poderia me recomendar um cafe amargo e suave

Digite a sua mensagem

Interface ChatBot

Chatbot da Serenatto

Chatbot

quais as características do grão bourbon vermelho

O Bourbon Vermelho é um dos nossos cafés mais populares!

As características do Bourbon Vermelho são:

- Doçura máxima (5/5): Ele tem uma doçura intensa que causa uma sensação aveludada no céu da boca.
- Corpo encorpado (5/5): O corpo do café é rico e cheio.
- Acidez baixa (1/5): A acidez é mínima, o que o torna uma opção ideal para quem busca um café suave.
- Amargor leve (2/5): O amargor é leve, o que não compromete a doçura do café.
- Notas de chocolate: O perfil sensorial do Bourbon Vermelho é marcado por notas de chocolate.

O processo de produção do Bourbon Vermelho é natural, cultivado em altitudes entre 1.200m e 1.450m, com torra média escura.

Você gostaria de saber mais sobre alguma outra variedade de café?

Digite a sua mensagem

Recursos

groqcloud

Playground

Chat Studio

llama-3.3-70b-versatile



</> View code

Playground

Documentation

Metrics

API Keys

Settings

Status

Discord

Chat with us



Personal

SYSTEM Enter system message (Optional)

PARAMETERS

Temperature

1

Max Completion Tokens

1024

Stream



JSON Mode



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



Recursos

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

groqcloud

Playground

Documentation

Metrics

API Keys

Settings

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

Status

Discord

Chat with us

Personal

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 groqcloud interface. On the left is a sidebar with various links: Playground, Documentation, Metrics, API Keys (which is currently selected and highlighted in red), Settings, Status, Discord, Chat with us, and Personal. The main content area is titled "API Keys" and contains a message: "Manage your API keys. Remember to keep your API keys safe to prevent unauthorized access." Below this is a table with columns: NAME, SECRET KEY, CREATED, LAST USED, and USAGE (24HRS). Two API keys are listed: "GROQ_API" and "groq_api_key". At the bottom left of the main area is a "Create API Key" button. A modal window titled "Create API Key" is open in the center. It has a text input field containing "api2" and a "Submit" button below it. At the bottom of the modal, there is an error message: "Falha! Esta tendo problemas? Send Feedback" next to a red exclamation mark icon. The Cloudflare logo and links for "Privacidade" and "Termos" are at the bottom right of the modal. The overall background is dark.

Recursos

Copiar a API gerada e salvar, pois precisaremos dela

The screenshot shows the groqcloud interface. On the left, there's a sidebar with links like Playground, Documentation, Metrics, API Keys (which is highlighted in red), Settings, Status, Discord, Chat with us, and Personal. The main area is titled "API Keys" and contains a message: "Manage your API keys. Remember to keep your API keys safe to prevent unauthorized access." Below this is a table with columns "NAME" and "SECRET". Three rows are listed: "GROQ_API" (secret gsk_...), "groq_api_key" (secret gsk_...), and "api2" (secret gsk_...). At the bottom of the table is a "Create API Key" button. A modal window is open over the table, titled "Create API Key". It contains the message "Your new API key has been created. **Copy it now, as we will not display it again.**" followed by the generated API key: `jsk_Z2KfYRPQQbNuRBeRKXZnWGdyb3FYrD8uNFbEBhQOEEnnmCKqTKrwO`. There is a "Copy" button next to the key. Below the key, there's a success message with a green checkmark: "Sucesso!" and a Cloudflare logo with links to "Privacidade" and "Termos". At the bottom right of the modal is a "Done" button.

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

