

6 - Prática: Embedding (II)

Vitor Manoel Emidio Machado

1- How to Get Started With Qdrant Cloud

Neste primeiro tópico nos é ensinado a como criar uma conta no Qdrant e começar a utilizá-lo, porém antes, para questões de esclarecimento, o Qdrant é um banco de dados vetorial otimizado para buscas de similaridade, sendo ideal para aplicações como: (Sistemas de recomendação, detecção de anomalias e dentre outros).

Após criarmos nossa conta, podemos produzir nosso primeiro cluster e gerar a nossa própria chave API, logo após, necessitamos executar o nosso SDKs para que possamos nos conectar aos servidores da Qdrant a iniciar os trabalhos.

```
</> bash  python  typescript  rust  java  csharp  go

from qdrant_client import QdrantClient

qdrant_client = QdrantClient(
    host="xyz-example.eu-central.aws.cloud.qdrant.io",
    api_key="<your-api-key>",
)
```

Código para inicialização do SDKs via Python

2.1- Using Lanflow with Qdrant Vector Store

Nessa próxima atividade nos é apresentado um exercício no qual utilizamos a Qdrant junto com o Lanflow, antes disso precisamos iniciar esse código, para que o Qdrant possa se conectar à instância do LangFlow, lembrando que a url se refere ao EndPoint do Qdrant e a chave Api a sua própria:

```
!pip install qdrant-client

from qdrant_client import QdrantClient

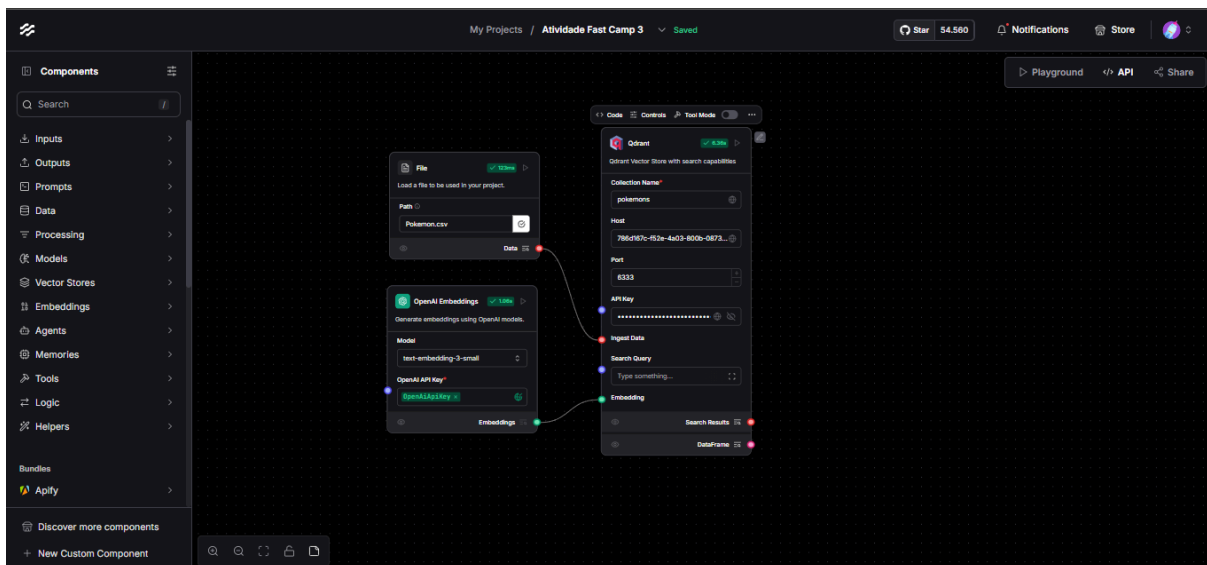
qdrant_client = QdrantClient(
    url="786d167c-f52e-4a03-800b-0873ed547f34.us-east4-b.gcp.cloud.qdrant.io",
    api_key="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhbnV2N1c3M1O1JtIn8.kdxQNYjZmZV8gIy90jWZX_Oo1CXjUN58p9EtuZqw",
)

print(qdrant_client.get_collections())
```

Logo após, iremos criar um fluxo dentro do Langflow, no qual terá o objetivo de inserir um arquivo Csv dentro das Collections do Qdrant.

Sendo essa composta por:

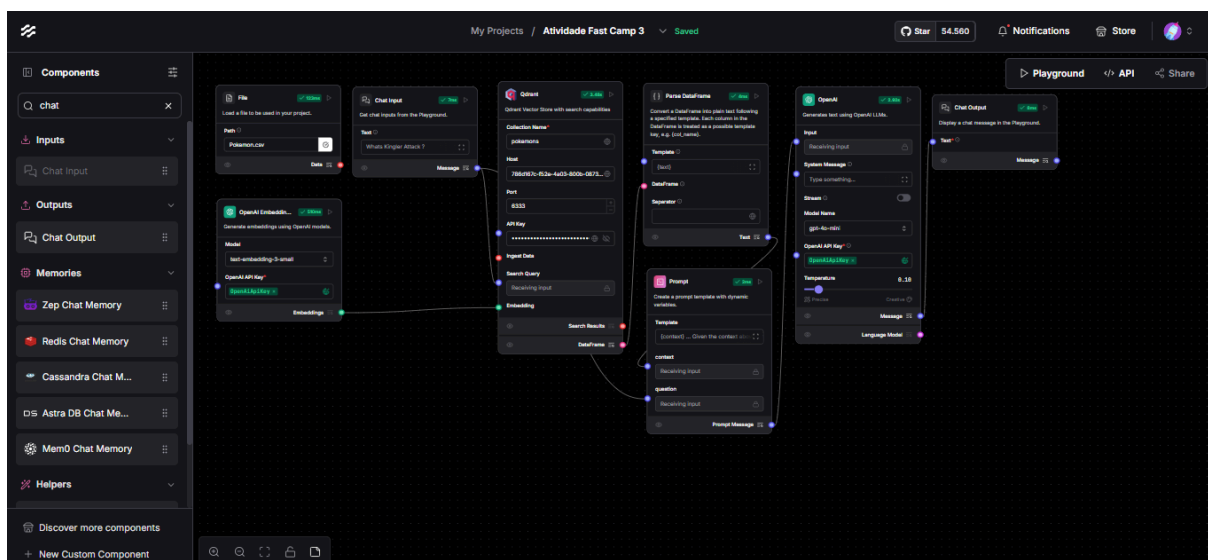
- File : Local aonde iremos carregar o arquivo Csv
- Api Embeddings : Transforma os dados em embeddings
- Qdrant : Banco vetorial onde os dados serão armazenados

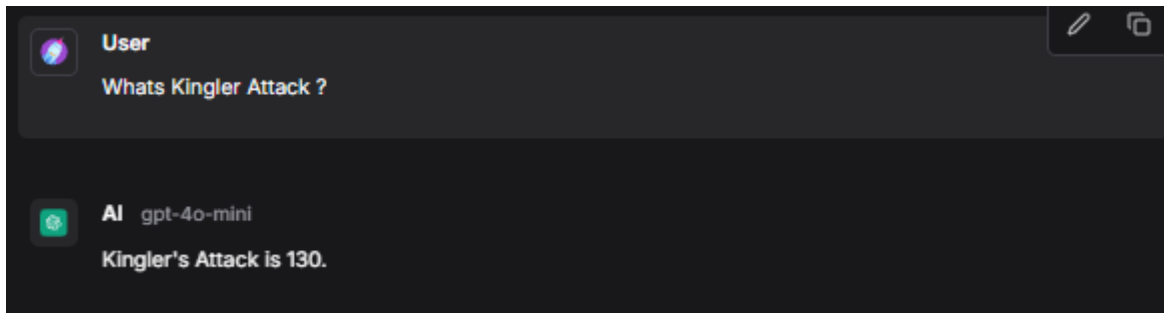


Após isso podemos ver o resultado na aba Collections do Dashboard do Qdrant:

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False	
2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False	
3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False	
3	VenusaurMega	Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	Charmander	Fire		309	39	52	43	60	50	65	1	False	

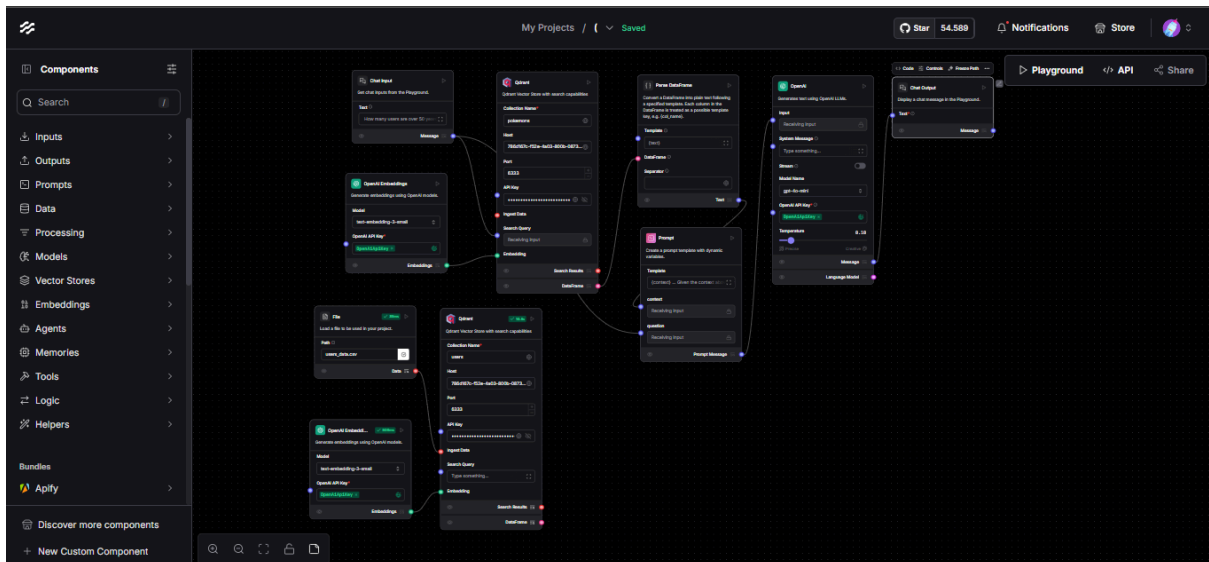
Agora para trabalhar com esses dados criamos um fluxo no qual iremos fazer uma pergunta sobre algum assunto (Pokemons no caso) e receberemos uma resposta baseados no Csv utilizado.

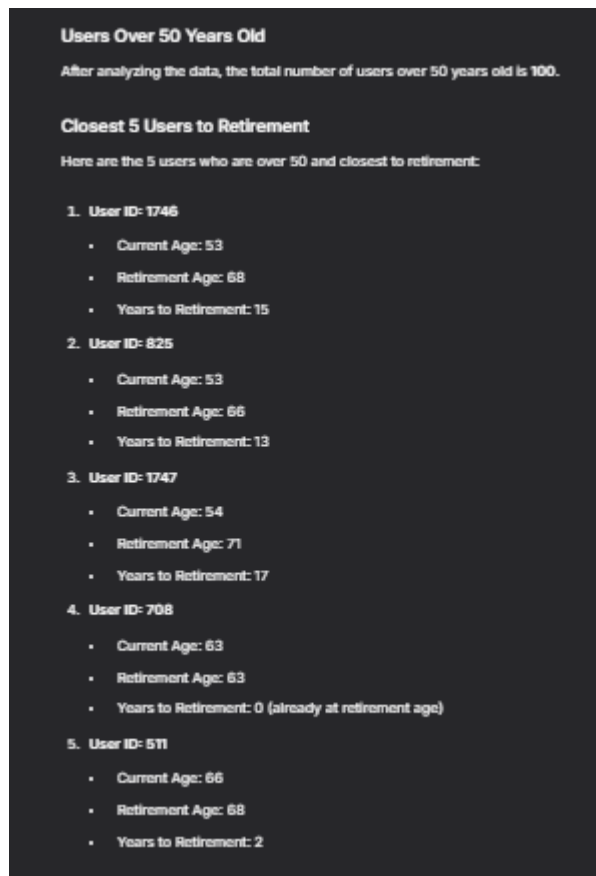




2.2- Prática

Para a minha prática eu segui os mesmo passos que o vídeo demonstra, porém nesse sentido eu utilizei um dataset do Kaggle, relacionado a transações financeiras (São dados reais porém as informações pessoais estão disfarçadas), o meu Csv chamado usuários possui users de um certo banco, meu objetivo era saber quantos possuíam idade superior a 50 anos, e quais eram as 5 pessoas que estavam mais próximas de se aposentar





3- Dificuldades

Nessa atividade não tive muitas dificuldades, apenas alguns erros no momento de integrar o Qdrant ao Langflow mas foi solucionado.

4- Conclusão

Posso concluir dessa atividade que, a utilização do Qdrant é um ferramenta extremamente poderosa para o fornecimento de informações ao Agent AI, abrindo várias possibilidades, como por exemplo um agente que faz análises financeiras, ou até mesmo um agente que trabalha com o estoque de um supermercado, podemos alimentar a AI com diversas informações para solucionar problemas do dia a dia e comerciais.

5- Referências

[Cloud Quickstart - Qdrant](#)

[Using Langflow with Qdrant Vector Store](#)

 [Financial Transactions Dataset: Analytics | Kaggle](#)