

# Uso de modelos de lenguaje de gran tamaño (LLM) de código abierto

*¿Es posible hacerlo "On Premise"?*

Isaac Vidal Daza

Apoyo a la Docencia  
Centro de Servicios de Informática y Redes de Comunicaciones  
Universidad de Granada

28-05-2024



**CSIRC**  
CENTRO DE SERVICIOS DE INFORMÁTICA Y REDES DE COMUNICACIONES



**UNIVERSIDAD  
DE GRANADA**

## Members

- Francisco Romera Juárez. (Head)
- Fernando López Álvarez.
- José Guerrero Peregrina.
- Antonio Cano Ruano.
- Rodrigo González Gálvez.
- Domingo Baca Ruíz.
- Leire Melchor López.
- Isaac Vidal Daza.

## Services

- Computers Classrooms Management.
- Software Deployment.
- Virtual Desktop Infrastructure.
- Microsoft 365 Management.
- Systems and Services Deployment.

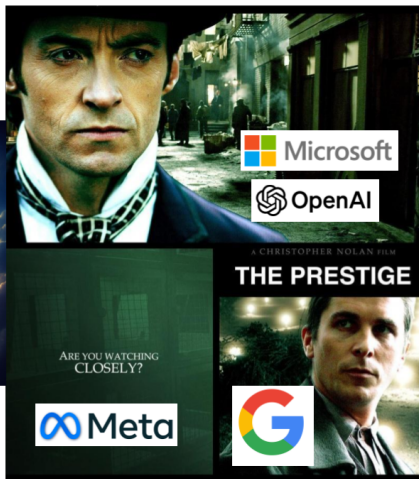
## Inventory

- 25 Faculties (Ceuta y Melilla).
- 128 Computers Classrooms.
- 3418 PC's.
- 200 Virtual Desktops.

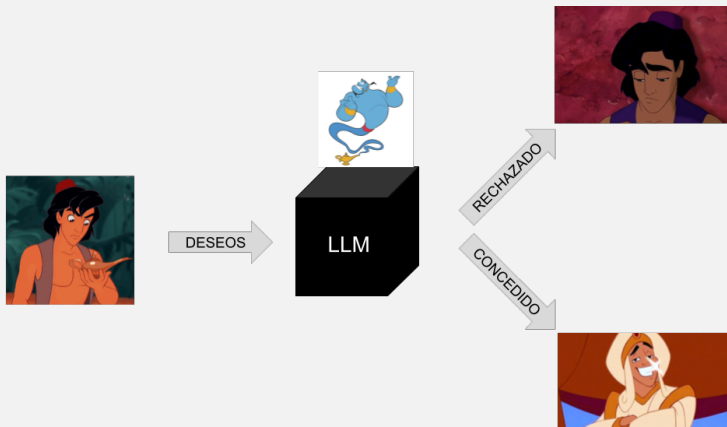
## Users

- 53000 Students.
- 3600 Teaching Staff.
- 2200 Administration Staff.

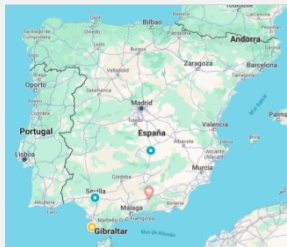
# Inteligencia Artificial (según los "Mass Media")



## Lámpara Mágica



## Instrucciones Precisas



# ¿Cómo pasar de Espectador a Ilusionista?

## Utilizando modelos LLM Open Source

- Llama3\*
- Mistral

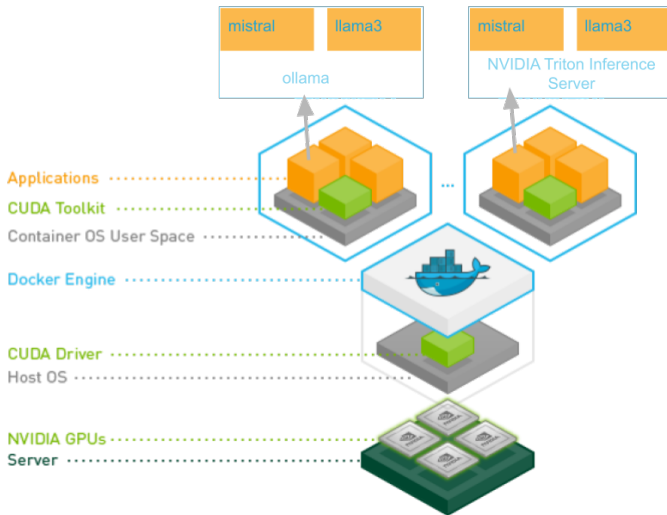
## ¿Dónde se encuentran?

- <https://huggingface.co/>
- <https://mistral.ai/>

## ¿Dónde ejecutan?

- On Premise
- Nube

# ¿Cómo ejecutarlos? CPUs y GPUs





# Interfaz Gráfica (unas 40 líneas de código)

## Listing 1: Código usando el framework Chainlit

```
1 @cl.on_chat_start
2 async def on_chat_start():
3     model = Ollama(base_url="http://localhost:11434", model="llama3")
4     prompt = ChatPromptTemplate.from_messages(
5         [
6             (
7                 "system",
8                 "You are a helpful assistant. You must always respond to
9                 Spanish questions if you receive a question in any other
10                language. "
11            ),
12            ("human", "{question}"),
13        ]
14    )
15    runnable = prompt | model | StrOutputParser()
16    cl.user_session.set("runnable", runnable)
```

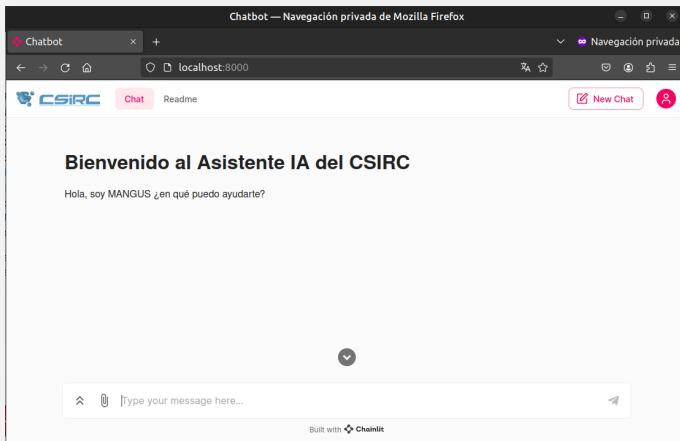
# Interfaz Gráfica (unas 40 líneas de código)

## Listing 2: Código usando el framework Chainlit

```
1 @cl.on_message
2 async def on_message(message: cl.Message):
3     runnable = cl.user_session.get("runnable") # type: Runnable
4
5     msg = cl.Message(content="")
6
7     async for chunk in runnable.astream(
8         {"question": message.content},
9         config=RunnableConfig(callbacks=[cl.LangchainCallbackHandler()]),
10    ):
11         await msg.stream_token(chunk)
12
13     await msg.send()
```

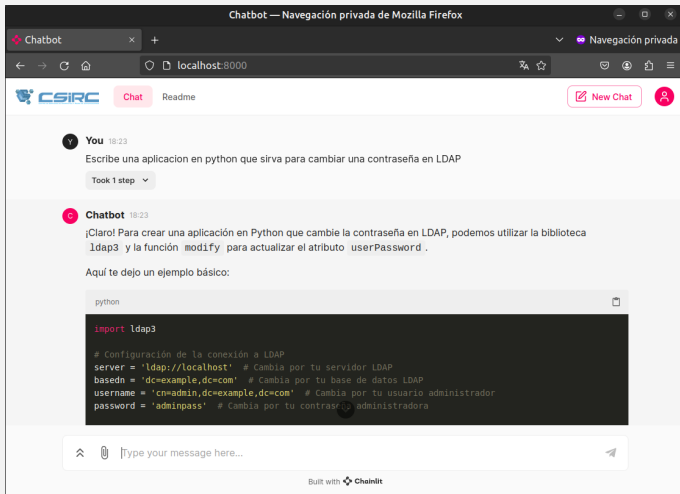
# Resultado 40 líneas de código

## Asistente Personalizado



# Resultado 40 líneas de código

## Asistente Personalizado



Chatbot — Navegación privada de Mozilla Firefox

Chatbot

localhost:8000

CSIRC Chat Readme New Chat

**You** 18:23

Escribe una aplicación en python que sirva para cambiar una contraseña en LDAP

Took 1 step

**Chatbot** 18:23

¡Claro! Para crear una aplicación en Python que cambie la contraseña en LDAP, podemos utilizar la biblioteca `ldap3` y la función `modify` para actualizar el atributo `userPassword`.

Aquí te dejo un ejemplo básico:

```
python

import ldap3

# Configuración de la conexión a LDAP
server = 'ldap://localhost' # Cambia por tu servidor LDAP
basedn = 'dc=example,dc=com' # Cambia por tu base de datos LDAP
username = 'cn=admin,dc=example,dc=com' # Cambia por tu usuario administrador
password = 'adminpass' # Cambia por tu contraseña administradora
```

⤴️ 📎 |Type your message here... ➡️

Built with Chainlit



# Resultado 40 líneas de código

## Asistente Personalizado

Chatbot — Navegación privada de Mozilla Firefox

Chatbot x + Navegación privada

localhost:8000

CSIRC Chat Readme New Chat

```
password = 'adminpass' # Cambia por tu contraseña administradora

# Conectamos a la base de datos LDAP
conn = ldap3.Connection(server, user=username, password=password, authentication='regular')

# Damos los detalles del usuario y la nueva contraseña
username_old = 'cn=user1,dc=example,dc=com'
new_password = 'nuevacontraseña123'

try:
    # Buscamos el usuario en LDAP
    entry = conn.search(search_scope='one', base=basedn, filter='(&(objectClass=person)(cn={}))'.format(username_old))

    # Actualizamos la contraseña del usuario
    conn.modify(entry[0].dn, {'userPassword': [(ldap3.MOD_REPLACE, new_password.encode('utf-8'))]})

    print('Contraseña actualizada con éxito!')
except Exception as e:
    print('Error:', str(e))
finally:
    conn.unbind()
```

Ten en cuenta que debes reemplazar los valores de `server`, `basedn`, `username` y `password` por tus

Type your message here...

Built with Chainlit



# ¿Puede tener memoria nuestro Asistente IA?

## ¿Base de Datos SQL?

animal	color	tags
cat	black, white	cute green eyes green background

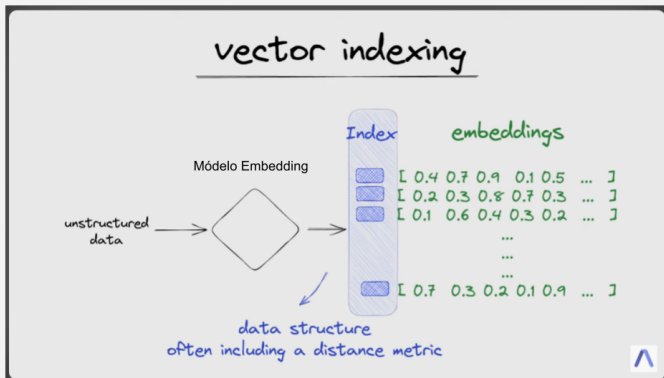
Full screen (f)

1:08 / 4:22 • Why do we need vector databases >

<https://www.youtube.com/watch?v=dN0lsF2cvm4>

# ¿Puede tener memoria nuestro Asistente IA?

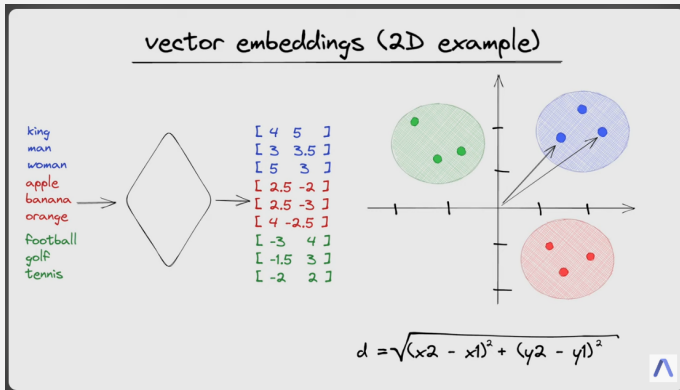
## Bases de Datos Vectoriales



<https://www.youtube.com/watch?v=dN0lsF2cvm4>

# ¿Puede tener memoria nuestro Asistente IA?

## Bases de Datos Vectoriales

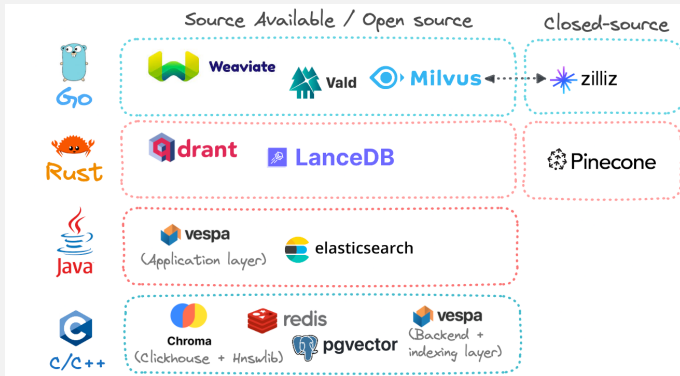


<https://www.youtube.com/watch?v=dN0lsF2cvm4>



# ¿Puede tener memoria nuestro Asistente IA?

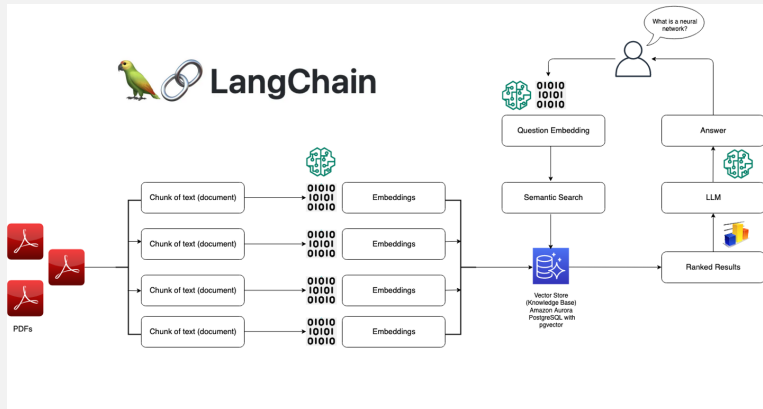
## Bases de Datos Vectoriales



<https://thedataquarry.com/posts/vector-db-1/>

# ¿Cómo utilizar una BD Vectorial con un LLM

## Frameworks LLM: Langchain y/o LlamaIndex



<https://www.langchain.com/langchain>

## Presentación y Código

<https://github.com/isvida/2024-RedIris>