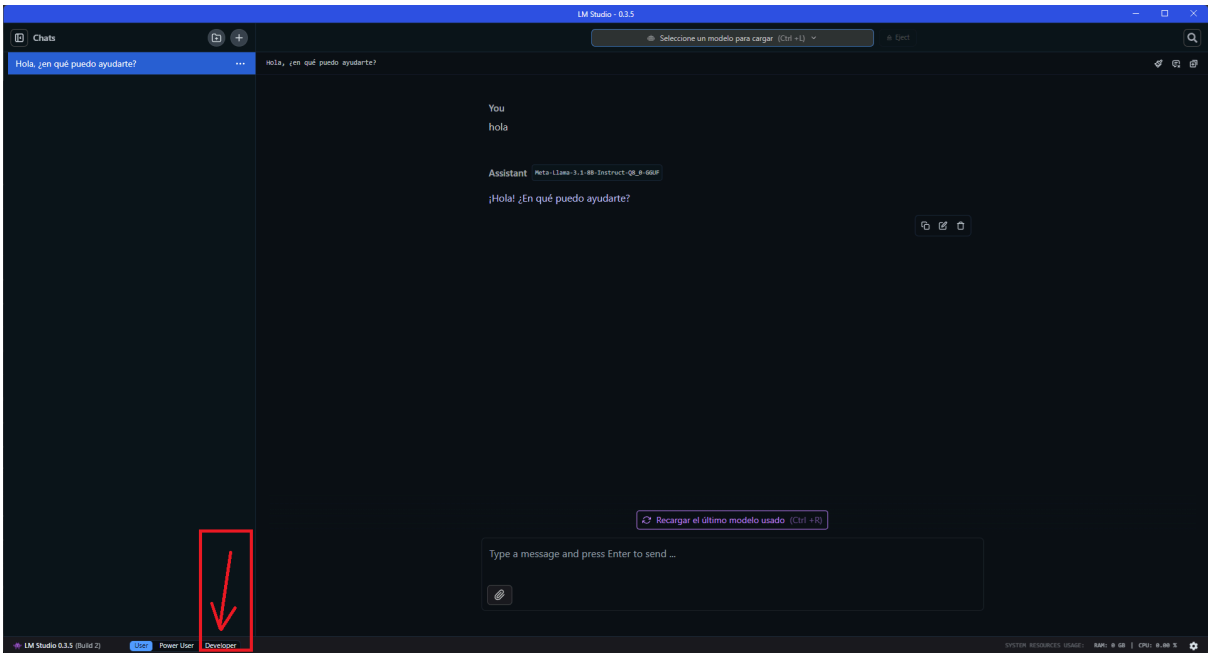
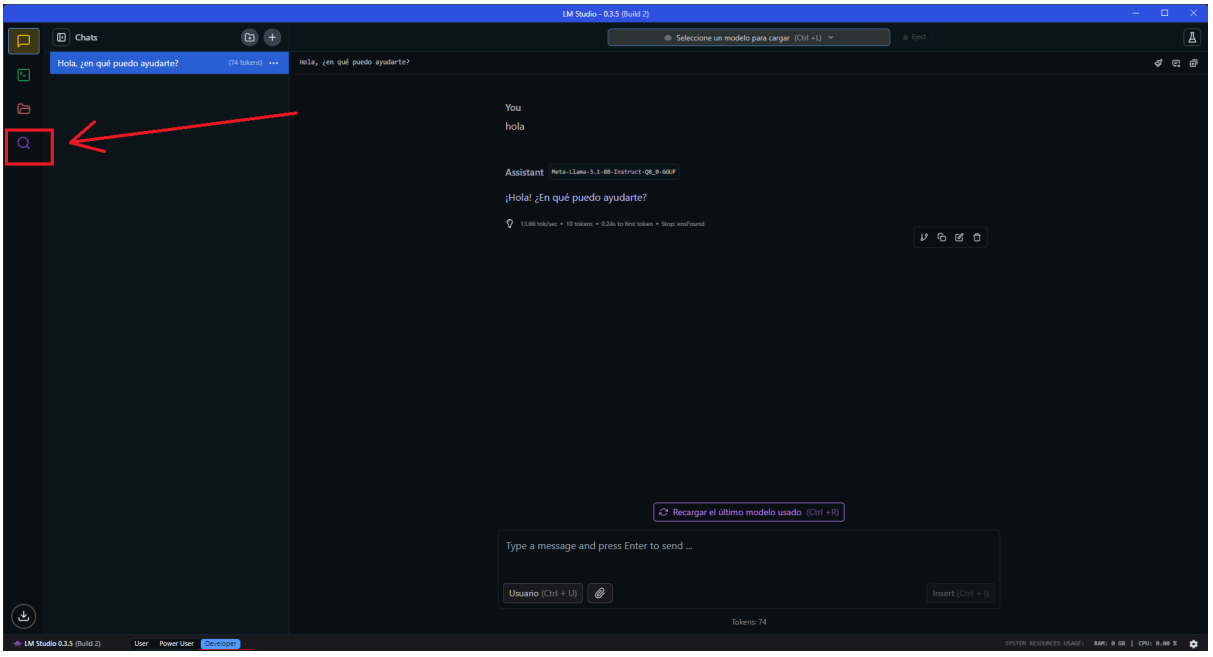
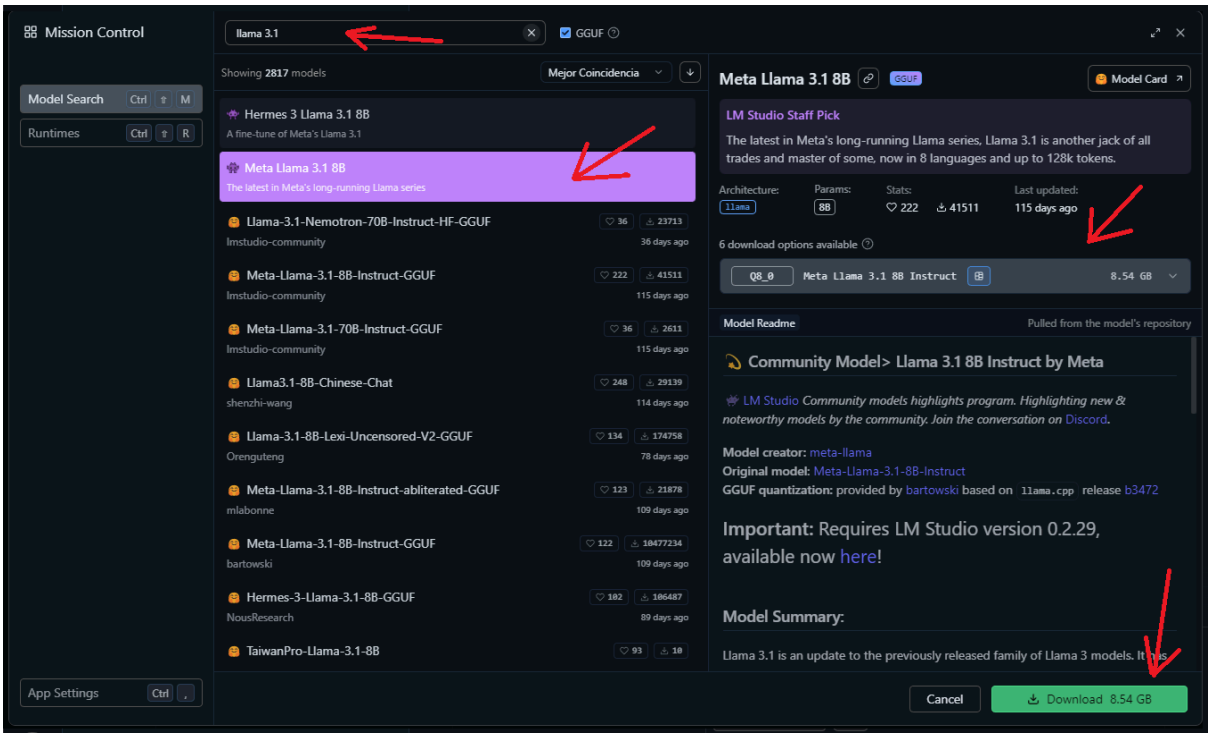
Paso 1: modo “developer”:



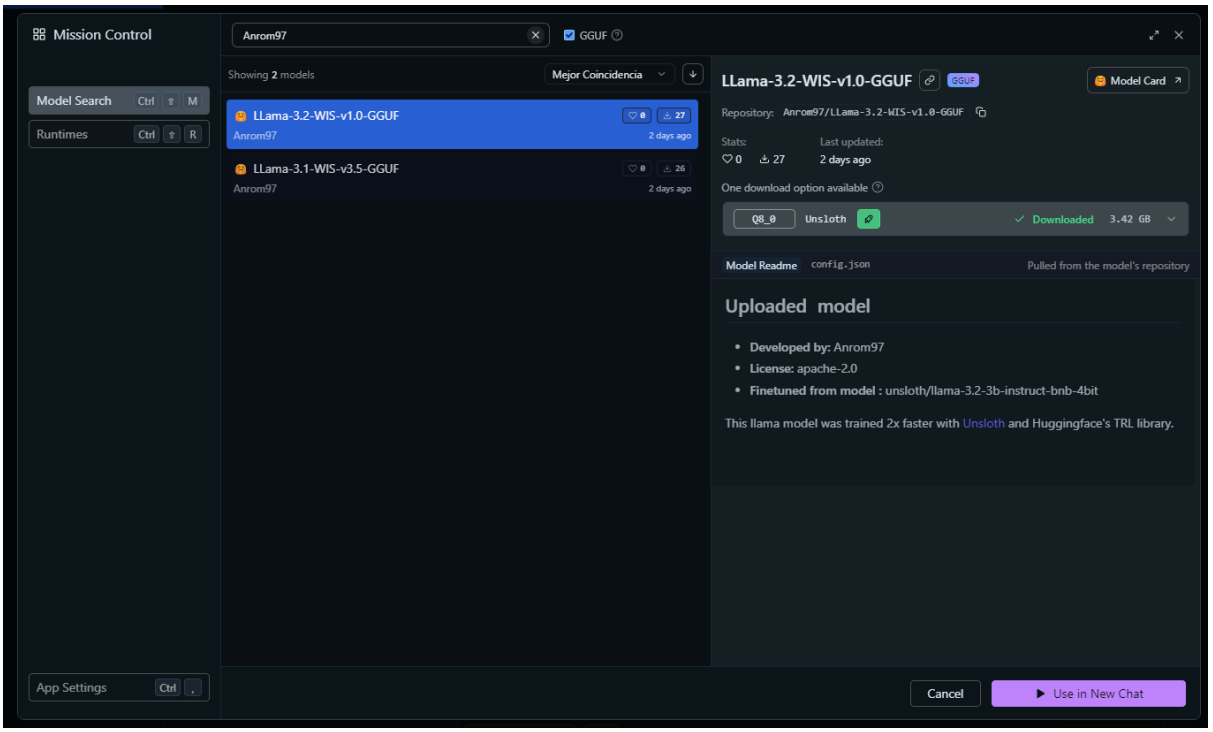
Paso 2: Buscamos un modelo para descargar.



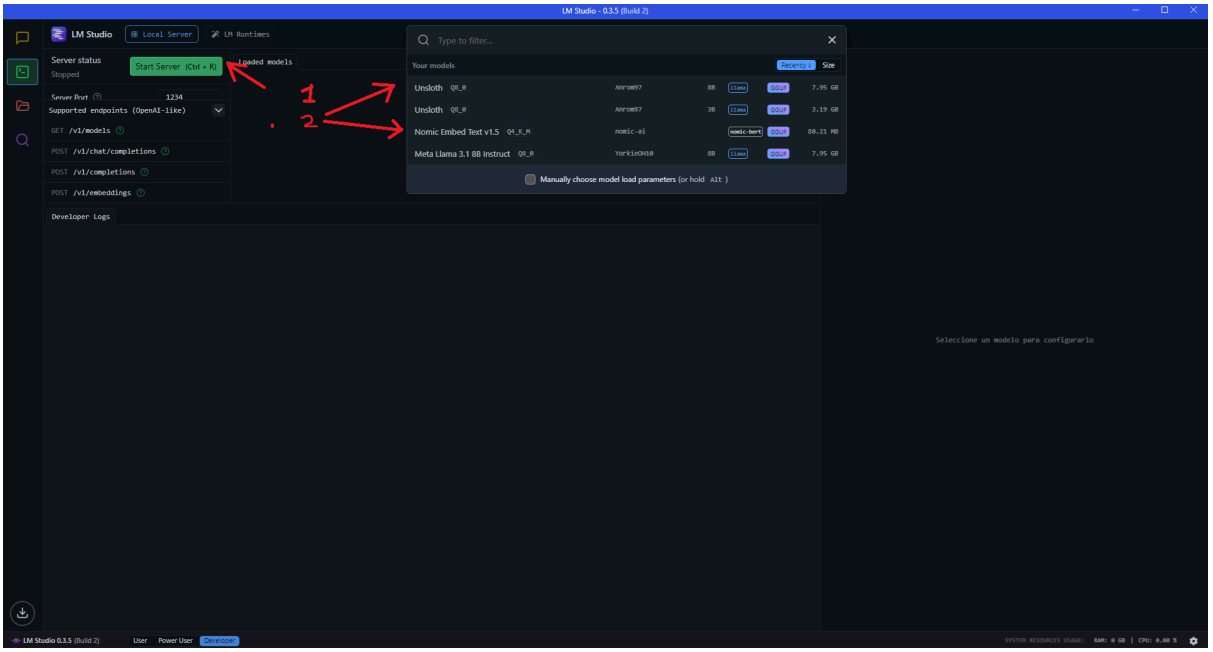
Paso 3: descargar un modelo



En nuestro caso tenemos nuestras propias versiones:



Paso 4: Levantar el servidor.



Primero: Cargamos el modelo (**llama 3.1 WIS**).
Segundo: Cargamos el modelo de embeding (Nomic Embed Text 1.5)
Tercero: Oprimimos en “**Start Server**”.

Correr el chatbot:

Pre requisitos: Cuda toolkit y PyThorch

Página oficial de Cuda toolkit: https://developer.nvidia.com/cuda-downloads?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exe_local

Página oficial de PyThorch: https://pytorch.org/get-started/locally/

Librerías necesarias (instalar antes de correr los scripts):

pip install numpy==1.26.4
pip install scikit-learn==1.5.2
pip install requests==2.32.3
pip install tqdm==4.66.5
pip install streamlit==1.38.0
pip install python-docx==1.1.2
pip install pypdf==5.1.0
pip install sentence-transformers==2.7.0
pip install chromadb==0.5.20
pip install llama-index-embeddings-huggingface
pip install llama-index-embeddings-instructor
pip install llama-index
pip install langchain==0.3.12
pip install langchain-community==0.3.12
pip install langchain-core==0.3.25
pip install langchain-huggingface==0.1.2
pip install langchain-chroma==0.1.4
pip install langchain-openai==0.1.14
pip install langchain-text-splitters==0.3.3
pip install langchainhub==0.1.21

También se puede instalar todas las dependencias de una sola vez utilizando el archivo requirements.txt: **pip install -r requirements.txt**

Crear la base de datos de vectores de RAG:

python create_database.py

Correr el chatbot (Es necesario tener el servidor de LLMStudio levantado):

streamlit run app.py