

# OPEN SOURCE VS CLOSED SOURCE

El Fascinante Mundo de los Modelos Open Source y Privados en IA Generativa

## Bloque 1: Closed Source en IA: Control y Seguridad

### Introducción:

En el ámbito de la IA generativa, los modelos Closed Source, también conocidos como modelos privados, son aquellos cuyo código fuente y parámetros internos no están disponibles al público. Los usuarios interactúan con estos modelos a través de interfaces (APIs) sin tener acceso a su funcionamiento interno.

### 1.1. Funcionamiento de los Modelos Closed Source

Para usar un modelo Closed Source, se adquiere una licencia, generalmente por tiempo limitado o por un número determinado de usos. Esta licencia nos permite acceder al software a través de una API (Application Programming Interface), que nos permite enviar peticiones al modelo y recibir sus respuestas.

No podemos ver cómo está programado el software ni realizar modificaciones. Estamos limitados a la funcionalidad y las configuraciones que la empresa proveedora nos ofrece.

### 1.2. Ejemplos de Modelos Closed Source

- **OpenAI: ChatGPT**

ChatGPT, desarrollado por OpenAI, es un modelo Closed Source de IA generativa ampliamente conocido y utilizado. Se interactúa con él a través de una API o de su interfaz web, sin tener acceso a su código fuente o a los pesos del modelo. OpenAI ofrece distintas versiones de ChatGPT (GPT-4, GPT-4 Mini, etc.) con diferentes capacidades y precios.

- **Anthropic: Claude**

Anthropic, otra empresa prominente en el campo de la IA Generativa, ha optado por el enfoque Closed Source con su modelo Claude. Este modelo se caracteriza por su enfoque en la seguridad y en ofrecer respuestas más coherentes y menos propensas a generar contenido problemático.

Claude se ofrece en distintos tamaños (Haiku, Sonnet y Opus), cada uno con capacidades y tiempos de respuesta diferentes, ajustándose a diversas necesidades.

### 1.3. Ventajas de los Modelos Closed Source en IA Generativa

- **Control y seguridad:** La empresa proveedora tiene control total sobre el modelo, lo que permite una mayor seguridad y protección de la propiedad intelectual.
- **Soporte técnico especializado:** Las empresas que desarrollan estos modelos ofrecen soporte técnico a los usuarios, lo que facilita su implementación y uso.

- **Actualizaciones y mejoras continuas:** Los modelos Closed Source suelen recibir actualizaciones y mejoras por parte de la empresa, lo que asegura un mejor rendimiento y nuevas funcionalidades.
- **Escalabilidad y rendimiento optimizados:** Las empresas que ofrecen estos modelos se encargan de la infraestructura necesaria para su ejecución, lo que asegura un alto rendimiento y una gran escalabilidad.
- **Facilidad de uso:** Las APIs y las interfaces web simplifican el uso de los modelos, incluso para usuarios sin conocimientos técnicos profundos.

#### 1.4. Desafíos de los Modelos Closed Source en IA Generativa

- **Falta de transparencia:** No podemos ver el código fuente ni los datos de entrenamiento, lo que dificulta la comprensión de cómo funciona el modelo y la evaluación de sus posibles sesgos.
- **Dependencia del proveedor:** Los usuarios dependen totalmente de la empresa proveedora para el acceso, mantenimiento y actualización del modelo.
- **Limitaciones en la personalización:** Las opciones de personalización suelen ser limitadas, ya que no podemos modificar el código fuente del modelo.
- **Costos:** Las licencias de acceso a estos modelos pueden ser costosas, especialmente para un uso intensivo o comercial.

#### 1.5. Resumen del Bloque 1

Los modelos Closed Source de IA generativa se caracterizan por su control, seguridad, soporte técnico y facilidad de uso. Sin embargo, presentan una falta de transparencia, dependencia del proveedor y limitaciones en la personalización, además de posibles costos asociados a su uso.

### Bloque 2: Open Source en IA: Flexibilidad y Colaboración

#### Introducción:

Como contrapartida a los modelos Closed Source, los modelos Open Source en IA generativa se caracterizan por la accesibilidad a su código fuente y a sus parámetros internos. Esto permite a los usuarios comprender su funcionamiento, realizar modificaciones e incluso contribuir a su desarrollo.

#### 2.1. La Revolución del Open Source en IA Generativa

Si bien hace unos años los modelos Open Source se consideraban menos potentes que sus contrapartes privadas, la situación ha cambiado radicalmente en los últimos tiempos. La liberación de modelos como Llama 2 de Meta ha marcado un punto de inflexión, ofreciendo un rendimiento comparable a los modelos Closed Source más avanzados.

## 2.2. Ejemplos de Modelos Open Source en IA Generativa

- **Meta: Llama 2**

Ya mencionado en el bloque anterior, Llama 2 es un ejemplo destacado de modelo Open Source en IA generativa. Disponible en varios tamaños, permite a los usuarios elegir el que mejor se adapte a sus necesidades, desde la experimentación en un ordenador personal hasta el desarrollo de aplicaciones a gran escala en infraestructuras en la nube.

- **Stability.AI: Stable Diffusion**

Stable Diffusion es un modelo Open Source para la generación de imágenes a partir de texto, similar en funcionalidad a DALL-E de OpenAI. Su liberación ha democratizado el acceso a esta tecnología, permitiendo a los usuarios crear imágenes sorprendentes sin necesidad de pagar por un servicio privado.

## 2.3. Ventajas de los Modelos Open Source en IA Generativa

- **Transparencia:** El acceso al código fuente permite comprender cómo funciona el modelo, identificar sus fortalezas, debilidades y posibles sesgos.
- **Personalización:** Podemos modificar el modelo para adaptarlo a nuestras necesidades específicas.
- **Colaboración:** La comunidad Open Source contribuye al desarrollo y mejora de los modelos, acelerando la innovación en el campo de la IA generativa.
- **Control:** Al ejecutar el modelo en nuestro propio hardware, tenemos control total sobre los datos utilizados y la privacidad de la información.
- **Costos:** Los modelos Open Source son gratuitos, lo que reduce significativamente los costos de desarrollo y experimentación.

## 2.4. Desafíos de los Modelos Open Source en IA Generativa

- **Complejidad:** Instalar, configurar y ejecutar un modelo Open Source puede requerir conocimientos técnicos avanzados.
- **Recursos:** Los modelos de gran tamaño pueden demandar un hardware potente para su ejecución, lo que puede suponer una barrera de entrada para algunos usuarios.
- **Mantenimiento:** Es necesario estar al día de las actualizaciones y realizar las modificaciones necesarias para mantener el modelo funcionando correctamente.

## 2.5. Resumen del Bloque 2

Los modelos Open Source de IA Generativa han experimentado un auge en los últimos tiempos, ofreciendo un rendimiento comparable a los modelos privados y brindando ventajas en transparencia, personalización, colaboración y control. Sin embargo, es importante tener en cuenta su complejidad, los recursos de hardware que pueden requerir y la necesidad de realizar un mantenimiento adecuado.

**Recordatorio Clave:** Al trabajar con modelos Open Source, es fundamental comprender su licencia y los permisos de uso, distribución y modificación que nos otorga.

BIG school

## Apuntes: Ejecución de Modelos Open Source en Local

### Introducción:

Como hemos visto anteriormente, el universo Open Source nos ofrece una gran variedad de modelos de IA generativa, con la ventaja de que podemos acceder a su código fuente y adaptarlos a nuestras necesidades. Para trabajar con estos modelos, es importante aprender a ejecutarlos localmente, es decir, en nuestro propio ordenador.

Si bien existen numerosas formas de ejecutar modelos Open Source en local, en este bloque nos centraremos en dos herramientas muy completas y sencillas de usar: NVIDIA ChatRTX y la plataforma Pinocchio.

### 2.6. NVIDIA ChatRTX: Un Chatbot Personalizado

ChatRTX es una aplicación de escritorio desarrollada por NVIDIA que nos permite ejecutar LLMs (Large Language Models) localmente, aprovechando la potencia de las GPUs NVIDIA RTX.

#### 2.6.1. Requisitos del Sistema:

Para utilizar ChatRTX, nuestro ordenador debe cumplir con los siguientes requisitos:

- **Sistema Operativo:** Windows 11
- **GPU:** NVIDIA GeForce RTX 30 o 40 Series GPU, o NVIDIA RTX A Series (Ampere o Ada Generation) con al menos 8GB de VRAM.
- **RAM:** 16GB o superior.

**Tip importante:** Sin una GPU compatible, ChatRTX no funcionará.

#### 2.6.2. Interfaz y Funcionalidades:

La interfaz de ChatRTX es muy sencilla e intuitiva. Al abrir la aplicación, veremos una ventana principal con tres secciones:

1. **Sección de Modelo:** En esta sección podemos seleccionar el modelo de IA que queremos usar. ChatRTX nos muestra información sobre cada modelo, incluyendo su tamaño, el idioma que soporta y si está descargado o no.
2. **Sección de Datos:** Aquí podemos elegir el tipo de datos con los que queremos interactuar. Tenemos tres opciones:
  - **Chat with all model data:** Interactuar con el modelo utilizando sus datos de entrenamiento originales.
  - **Chat with your data (RAG):** Cargar nuestros propios documentos (txt, pdf, docx) y hacer preguntas al modelo sobre su contenido. Esta funcionalidad se basa en la técnica RAG (Retrieval-Augmented Generation).

- **Chat with the dataset the AI model trained on:** Interactuar con el modelo utilizando el mismo dataset con el que fue entrenado.
3. **Sección de Chat:** En esta sección, podemos escribir nuestras preguntas al modelo y ver sus respuestas.

### 2.6.3. Ejemplo de Uso:

Para empezar a usar ChatRTX, seleccionamos un modelo descargado (por ejemplo, Mistral 7B int4), elegimos la fuente de datos y empezamos a interactuar con el modelo escribiendo nuestras preguntas en la sección de chat. ChatRTX nos permitirá conversar con el modelo, solicitarle información, resúmenes, traducciones, etc.

## 2.7. Pinocchio: Instalar y Ejecutar Aplicaciones de IA con un Clic

Pinocchio es una plataforma que facilita la instalación y ejecución de aplicaciones de IA en nuestro ordenador local. Funciona como un "navegador de aplicaciones", permitiéndonos instalar y controlar aplicaciones de IA con un solo clic.

### 2.7.1. Funcionalidades:

Pinocchio nos ofrece:

- **Instalación simplificada:** Instala automáticamente las aplicaciones de IA, incluyendo las dependencias necesarias.
- **Control de aplicaciones:** Permite gestionar las aplicaciones instaladas, iniciarlas, detenerlas y controlar su funcionamiento.
- **Repositorio de aplicaciones:** Cuenta con un extenso repositorio de aplicaciones de IA de diversos tipos, como procesamiento de imágenes, audio, video, LLMs y mucho más.

### 2.7.2. Ejemplo de Uso:

Para utilizar Pinocchio, simplemente descargamos e instalamos la aplicación. Al abrirla, veremos una ventana principal con dos secciones:

1. **"Verified Scripts":** Muestra aplicaciones de IA verificadas por Pinocchio.
2. **"Community Scripts":** Muestra aplicaciones de IA creadas por la comunidad.

Para instalar una aplicación, simplemente la seleccionamos y pulsamos el botón "Download". Pinocchio se encargará de descargarla e instalarla automáticamente en nuestro ordenador.

### 2.7.3. Ejemplo: Forge (Stable Diffusion en Local)

Forge es una de las aplicaciones de Pinocchio más populares, que nos permite ejecutar el modelo Open Source de generación de imágenes Stable Diffusion en nuestro ordenador local. Al instalarla a través de Pinocchio, podremos crear imágenes realistas a partir de texto, sin necesidad de utilizar servicios en la nube.

## 2.8. Resumen del Bloque 2

- NVIDIA ChatRTX y Pinocchio son dos herramientas que facilitan la ejecución de modelos Open Source de IA generativa en local.
- ChatRTX nos permite interactuar con LLMs a través de una interfaz de chat, mientras que Pinocchio nos permite instalar y controlar diversas aplicaciones de IA, incluyendo Stable Diffusion para la generación de imágenes.
- Es importante verificar los requisitos del sistema antes de instalar estas herramientas, ya que generalmente requieren hardware potente, como GPUs dedicadas.

### Recursos adicionales:

- **NVIDIA ChatRTX:** <https://www.nvidia.com/en-us/studio/chatrtx/>
- **Pinocchio:** <https://pinokio.computer/>

Apuntes: Google Colaboratory: Entorno Interactivo para IA

### Introducción:

Google Colaboratory, también conocido como Colab, es una poderosa herramienta gratuita que nos permite programar y ejecutar código Python en nuestro navegador, ideal para tareas de Machine Learning e IA generativa. Colab nos ofrece un entorno interactivo similar a Jupyter Notebook, con la ventaja de que no requiere configuración adicional, proporciona acceso a GPUs sin costo y facilita la colaboración entre usuarios.

### 3.1. Interfaz de Google Colaboratory

Al acceder a un notebook de Colab, nos encontramos con una interfaz similar a la siguiente:

#### Esquema 1: Interfaz de Google Colaboratory

- **Barra de Menú:** Contiene opciones para Archivo, Editar, Ver, Insertar, Entorno de ejecución, Herramientas, Ayuda.
- **Barra de Herramientas:** Contiene botones para ejecutar celdas, añadir celdas de código o de texto, subir archivos, conectar con un entorno de ejecución, compartir, etc.
- **Panel Lateral Izquierdo:**
  - **Índice:** Muestra los títulos y secciones del notebook para facilitar la navegación.
  - **Buscar y Reemplazar:** Permite buscar y reemplazar texto en el código.
  - **Variables:** Muestra las variables definidas en el notebook y sus valores.
  - **Archivos:** Permite acceder a los archivos almacenados en Google Drive.
- **Área Principal:** Contiene las celdas de código y de texto que componen el notebook.

### 3.2. Primeros Pasos con Colab

El documento que se abre al acceder a Colab es un entorno interactivo similar a un cuaderno donde podemos escribir y ejecutar código Python. Para ejecutar una celda de código, solo hay que hacer clic en ella y pulsar el botón de reproducción situado a la izquierda del código, o usar la combinación de teclas "Comando/Ctrl + Enter". Las variables que definamos en una celda se pueden usar después en otras celdas.

### 3.3. Combinando Código Ejecutable y Texto Enriquecido

Colab nos permite combinar código ejecutable y texto enriquecido en un mismo documento, además de imágenes, HTML, LaTeX y mucho más. Los cuadernos se almacenan en nuestra cuenta de Google Drive. Podemos compartirlos con nuestros colaboradores de forma que lo vean, lo modifiquen y lo ejecuten.

### 3.4. Ciencia de Datos con Colab

Colab es una herramienta ideal para proyectos de Ciencia de Datos, ya que proporciona acceso a las bibliotecas más populares de Python para analizar y visualizar datos. Por ejemplo, podemos utilizar NumPy para generar arrays y datos aleatorios y Matplotlib para visualizaciones.

### 3.5. Novedades: La API de Gemini

Colab ha incorporado la API de Gemini, el nuevo modelo de lenguaje de Google, para que podamos utilizarlo en nuestros notebooks.

### 3.6. Ventajas de Google Colaboratory

- No requiere configuración adicional.
- Acceso a GPUs sin costo adicional.
- Permite compartir contenido fácilmente.
- Colab puede facilitar tu trabajo, ya seas estudiante, científico de datos o investigador de IA.

#### Recursos adicionales:

- **Introducción a Colab:** <https://colab.research.google.com/>
- **Documentación de Colab:** [https://colab.research.google.com/github/googlecolab/colabtools/blob/master/googlecolab/tutorials/intro\\_to\\_colab.ipynb](https://colab.research.google.com/github/googlecolab/colabtools/blob/master/googlecolab/tutorials/intro_to_colab.ipynb)

**Recordatorio Clave:** Para un mejor rendimiento, es recomendable utilizar una cuenta de Google para acceder a Colab y aprovechar sus funcionalidades de almacenamiento en la nube y colaboración.