

Conversación con Gemini

Exported: 12/6/2026 12:12:36

Jordi H.

Has dicho cual es la url para usar la api de vllm?

Gemini

Para usar la API de **vLLM** (que es compatible con la especificación de OpenAI), la URL depende de cómo hayas configurado el host y el puerto al lanzar el servidor, pero por defecto es la siguiente

URL por defecto

- Base URL: `http://localhost:8000/v1`
- Endpoint de chat: `http://localhost:8000/v1/chat/completions`
- Endpoint de completado: `http://localhost:8000/v1/completions`

Cómo configurar la URL

Si estás lanzando el servidor desde la terminal, puedes personalizar la dirección y el puerto mediante los parámetros `--host` y `--port`. Por ejemplo:Bas

CODE

```
python -m vllm.entrypoints.openai.api_server \
--model mistralai/Mistral-7B-Instruct-v0.3 \
--host 0.0.0.0 \
--port 8080
```

En este caso, la URL pasaría a ser: `http://<tu-ip>:8080/v1`.

Ejemplos de uso

1. Con `curl`

Para verificar que el servidor responde correctamente: Bash

CODE

```
curl http://localhost:8000/v1/models
```

2. Con la librería de OpenAI en Python

Dado que vLLM imita la API de OpenAI, puedes usar su cliente oficial simplemente cambiando la `base_url`:Pytho

CODE

```
from openai import OpenAI

# No necesitas una API Key real a menos que la hayas configurado en el servidor
client = OpenAI(
    base_url="http://localhost:8000/v1",
    api_key="token-si-es-necesario"
)

completion = client.chat.completions.create(
    model="nombre-de-tu-modelo",
    messages=[{"role": "user", "content": "Hola, ¿cómo funciona vLLM?"}]
)

print(completion.choices[0].message.content)
```

Nota: Si has desplegado vLLM en un servidor remoto o dentro de un contenedor Docker/Podman (como sueles hacer con tus proyectos), asegúrate de mapear correctamente los puertos y usar la IP del servidor o el nombre del servicio en lugar de localhost

Jordi H.

Has dicho qué debo poner en la config de anythingllm?

Gemini

Para configurar **AnythingLLM** utilizando **vLLM** como motor (back-end), debes dirigirte a la sección de **Settings** > seguir estos pasos:

1. Selección del Proveedor

En el menú desplegable de **LLM Provider**, selecciona **Generic OpenAI**.

- Nota: Aunque vLLM no es OpenAI, su API es 100% compatible con su estándar, y esta es la forma más directa de conectarlos

2. Parámetros de Configuración

Verás un formulario con los siguientes campos que debes completar:

Campo	Valor / Instrucción
Base URL	http://localhost:8000/v1
API Key	Puedes dejarlo vacío o poner cualquier cadena (ej.
Chat Model Name	El ID exacto del modelo que cargaste en vLLM (ej:
Token limit	Depende del modelo (ej: 4096, 8192 o 32768).

Importante: Si AnythingLLM y vLLM están en contenedores distintos (por ejemplo, usando Podman o Docker), no uses localhost. Debes usar la IP de la máquina host o el nombre del contenedor dentro de la red compartida (ej: http://172.17.0.1:8000/v1)

3. Configuración del Embedding (Opcional)

AnythingLLM necesita un modelo de "Embedding" para procesar tus documentos locales. Tienes dos opciones

principales

1. AnythingLLM Default: Es el más sencillo, se ejecuta localmente dentro de la propia app.
2. vLLM (si el modelo lo soporta): Si has cargado un modelo que admite embeddings en vLLM, puedes repetir el proceso en la pestaña Embedding Settings usando la misma URL

4. Verificación

Una vez guardes los cambios, intenta crear un **Workspace**. Si la conexión es correcta, AnythingLLM te permitirá seleccionar el modelo de vLLM en la configuración del chat del workspace. Si recibes un error de conexión, verifica con un curl desde la terminal donde corre AnythingLLM para asegurar que el puerto 8000 es accesible:

CODE

```
curl http://localhost:8000/v1/models
```

¿Estás corriendo vLLM y AnythingLLM dentro del mismo entorno de red o están en máquinas diferentes?

Jordi H.

Has dicho puede open notebook usar un servidor vllm ?

Gemini

Sí, **Open Notebook** puede utilizar perfectamente un servidor **vLLM**. Al igual que AnythingLLM, Open Notebook está diseñado para ser agnóstico respecto al modelo y soporta proveedores que utilicen el estándar de la API de OpenAI. Para configurarlo, debes seguir estos pasos en la interfaz de Open Notebook

1. Agregar la credencial

1. Entra en Settings (Ajustes) y busca la sección API Keys.
2. Haz clic en Add Credential.
3. En el desplegable de Provider, selecciona OpenAI-Compatible.

2. Configurar la conexión

Rellena los campos con la información de tu servidor vLLM:

- Name: Un nombre para identificarlo (ej. vLLM-Local o Servidor-Laboratorio).
- Base URL: La dirección de tu servidor. Si Open Notebook corre en el mismo host que vLLM, usa `http://localhost:8000/v1`. Si están en contenedores distintos dentro de la misma red de Docker/Podman, usa la IP interna o el nombre del servicio
- API Key: vLLM no requiere una por defecto, pero Open Notebook puede pedirte que rellenes el campo. Puedes poner algo genérico como `vllm-token` o simplemente `none`

3. Descubrir y Registrar el modelo

Una vez guardada la credencial:

1. Haz clic en Test Connection para verificar que Open Notebook llega al servidor vLLM.
2. Haz clic en Discover Models. Open Notebook consultará el endpoint `/v1/models` de tu vLLM y listará los modelos que tienes cargados (por ejemplo, el que estés usando para tu proyecto de arquitectura de sistemas)
3. Selecciona el modelo que te interese y dale a Register Models.

Ventajas de usar vLLM con Open Notebook

Dado que mencionaste que trabajas con **arquitectura de sistemas** y **analítica de datos**, esta combinación es ideal por varias razones

- Rendimiento: A diferencia de Ollama, vLLM utiliza PagedAttention, lo que te permitirá procesar múltiples peticiones o documentos largos dentro de Open Notebook con una latencia mucho menor
- Privacidad: Mantienes todo el flujo RAG (Generación Aumentada por Recuperación) dentro de tu infraestructura o homelab, sin enviar datos de investigación a la nube
- Software Libre: Ambos proyectos siguen la filosofía de código abierto que priorizas.

¿Tienes pensado desplegar Open Notebook también mediante un contenedor Podman en tu servidor "ymir"?

Jordi H.

Has dicho hay version para windows de open notebook

Gemini

No existe una versión de **Open Notebook** como un instalable .exe tradicional para Windows, ya que está diseñado principalmente como una aplicación web auto-alojada. Sin embargo, puedes ejecutarlo en Windows de forma muy sencilla siguiendo estos métodos, siendo el primero el más recomendado para tu perfil técnico

1. Docker Desktop (Recomendado)

Es la forma oficial y más estable. Al usar contenedores, te aseguras de que todas las dependencias (como la base de datos SurrealDB que usa internamente) funcionen correctamente sin ensuciar tu sistema operativo

- Requisito: Instalar Docker Desktop con el backend de WSL 2.
- Pasos rápidos: Descarga el archivo docker-compose.yml desde su repositorio de GitHub. Ejecuta `docker compose up -d` en tu terminal (PowerShell o CMD). Accede a través de tu navegador en `http://localhost:8502`

2. Instalación desde el Código Fuente

Si prefieres no usar Docker, puedes clonar el repositorio y ejecutarlo usando **Python** y **Node.js**.

- Necesitarás instalar las dependencias del backend (FastAPI) y el frontend (Next.js).
- Ten en cuenta que también deberás gestionar manualmente la instancia de la base de datos SurrealDB.

3. Alternativas en la Microsoft Store

Existen aplicaciones con nombres similares en la tienda de Windows (como "Notebook - AI Note Taker"), pero **no son el mismo proyecto** de código abierto open-notebook.ai. Muchas de estas son aplicaciones cerradas o con suscripciones, por lo que te sugiero evitarlas si buscas la experiencia de software libre y control total. **Consejo para tu infraestructura**

Dado que ya estás trabajando con **vLLM** y tienes un servidor de laboratorio, lo más eficiente sería desplegar Open Notebook directamente en tu servidor Linux (o en tu Homelab) mediante Docker/Podman. De esta forma, podrías acceder a él desde cualquier equipo de tu red (incluyendo tu PC con Windows) simplemente entrando en la IP del servidor, sin tener que consumir recursos locales de tu estación de trabajo. ¿Te interesaría que te ayudara con el archivo docker-compose.yml para conectar ambos servicios en la misma

Jordi H.

Has dicho tengo un conjunto de documentos (word y excel) que describen los perfiles de trabajo (tareas, responsabilidades, skills, etc.) de una gran corporación. debo analizarlos en su conjunto y mostrar propuesta de similitud, unificación, corrección de disonancias, etc. debo usar un motor de ia local y herramientas open-source. A priori me gustaría que me expliques qué tipo de herramienta es la más adecuada: si es suficiente con una interfaz tipo anythingLLM que tiene rag y capacidad de gestionar documentos o otra tipo OpenNotebook que emula, creo, el notebooklm de google. qué ventajas me aportaría esta segunda op

Gemini

Para un proyecto de análisis corporativo que implica no solo **consultar** información (RAG), sino **encontrar** unificar criterios y detectar contradicciones**, la elección de la herramienta es clave. Aquí tienes la comparativa técnica para ayudarte a decidir qué se adapta mejor a tu rol de arquitecto de datos:

1. AnythingLLM (El enfoque RAG Tradicional)

AnythingLLM es excelente como "bibotecario inteligente". Su fuerza reside en la gestión de la base de datos vectorial

- **Cómo funciona:** Indexa los Word/Excel en una base de datos de vectores (como LanceDB o Chroma). Cuando haces una pregunta, busca los fragmentos más relevantes
- **Ventajas:** Es muy rápido para responder "qué tareas tiene el perfil X" o "busca quién tiene la skill Python".
- **Limitación para tu caso:** El RAG tradicional tiene dificultades con el análisis holístico. Si le pides "unifica todos los perfiles", el sistema solo recuperará unos pocos fragmentos (chunks) de cada documento para no exceder la ventana de contexto, lo que puede hacer que pierda la visión de conjunto o ignore disonancias sutiles entre departamentos

2. OpenNotebook / NotebookLM (El enfoque "Context-First")

Estas herramientas están diseñadas para la **síntesis y el razonamiento sobre un corpus cerrado**. A diferencia de un chat convencional, priorizan el análisis global del conjunto de fuentes

Ventajas específicas para tu análisis de perfiles corporativos:

- **Capa de Razonamiento sobre el Corpus:** Mientras que AnythingLLM busca "agujas en un pajar", OpenNotebook está diseñado para "describir el pajar". Es capaz de comparar fuentes de forma nativa para generar guías de estudio, resúmenes comparativos o FAQs automáticas
- **Detección de Disonancias:** Al tener una interfaz orientada a notas y cuadernos, facilita mucho más la tarea de "unificación". Puedes pedirle: "Compara las responsabilidades del Senior Data Architect en el departamento de IT vs. el de Operaciones y lista las contradicciones"
- **Citación y Trazabilidad:** Al emular a NotebookLM, suelen tener una gestión de citas más visual. Cuando te proponga una unificación, podrás ver exactamente de qué documentos (Word/Excel) ha sacado cada skill o tarea, algo vital para la validación humana en una gran corporación
- **Perspectivas automáticas:** Estas herramientas suelen ofrecer funciones de "brainstorming" o "crítica" basadas exclusivamente en tus documentos, lo cual es perfecto para tu objetivo de "propuestas de similitud"

¿Cuál es la más adecuada para ti?

Mi recomendación técnica: Dada la naturaleza de tu tarea (unificar y corregir), **OpenNotebook (o una Research Notebook)** es superior. **El análisis de perfiles profesionales requiere** evaluar semántica y jerarquía**, no solo buscar palabras clave. OpenNotebook te permitirá crear un "Cuaderno de Unificación" donde los documentos no son solo una base de datos, sino el contexto vivo de la conversación.

Estrategia de Implementación Pro (Aprovechando tu servidor Ymir)

Como vas a usar un motor local (vLLM):

1. Modelo: Para detectar "disonancias" y "similitudes" sutiles en textos corporativos, te sugiero un modelo con gran capacidad de razonamiento y ventana de contexto (mínimo 32k tokens), como Llama 3.1 8B o Mistral Nemo 12
2. Pipeline:
 - Usa OpenNotebook como interfaz de usuario para el análisis y la toma de notas.
 - Conéctalo a tu servidor vLLM (que ya tienes configurado) para que el procesamiento sea pesado y local.
 - Si los Excel son muy grandes o complejos (muchas columnas de skills), quizás necesites pre-procesarlos a Markdown para que la IA entienda mejor la estructura de filas y columnas antes de subirlos

Resumen: AnythingLLM es para preguntar a los documentos; OpenNotebook es para trabajar con los documentos. Para unificar perfiles de una corporación, necesitas **trabajar** con ellos

Jordi H.

Has dicho hay más herramientas tipo "ai research notebook", además de la open y la de google??

Gemini

Además de **Open Notebook** y la opción de Google, el ecosistema de herramientas de "AI Research Notebook" (especialmente las orientadas a la privacidad y el uso local) ha crecido mucho. Dado que buscas algo **open-source** y que puedas conectar a tu vLLM, estas son las alternativas más sólidas para tu análisis de perfiles corporativos

1. Khoj (El "Asistente de Investigación" más completo)

Es probablemente el competidor más fuerte de código abierto para NotebookLM. No es solo un chat, es una base de conocimiento semántica

- Por qué te sirve: Está diseñado específicamente para conectar tus archivos (PDF, Markdown, Word) y crear un "cerebro" digital
- Conexión con vLLM: Es totalmente compatible con APIs de OpenAI, por lo que puedes apuntarlo a tu servidor local
- Ventaja clave: Tiene una función de "Agentes" que podrías configurar específicamente para tu tarea (ej. un agente especializado en "Detección de incongruencias en RRHH")
- Instalación: Muy sencilla mediante Docker.

2. SurfSense (Enfoque en Investigación y Agentes)

Una herramienta emergente que se define como un "AI Research Agent" auto-alojado.

- Por qué te sirve: Mientras que otras herramientas son pasivas, SurfSense intenta ser más activa en la búsqueda de conexiones entre tus datos
- Ventaja clave: Permite citar fuentes de manera muy precisa, algo fundamental cuando tienes que justificar ante una corporación por qué propones unificar dos perfiles de trabajo
- Uso local: Funciona perfectamente con motores locales mediante Docker.

3. InsightsLM / OpenBookLM

Son proyectos que nacieron específicamente como "clones" abiertos de la interfaz de Google.

- OpenBookLM: Es ideal si buscas una experiencia casi idéntica a la de Google pero en tu propio hardware. Se centra mucho en la trazabilidad: cada respuesta tiene un enlace directo al fragmento del documento original

- InsightsLM: Muy similar, prioriza que la IA esté "blindada" (grounded) a tus documentos, evitando que alucine con información externa

Tabla Comparativa para tu proyecto de Perfiles Profesionales

Herramienta	Capacidad de Síntesis	Control de Fuentes	Complejidad de Setup
Open Notebook	Muy Alta (Notebook style)	Excelente	Media (Docker)
Khoj	Alta (Asistente personal)	Muy buena	Media
AnythingLLM	Media (RAG puro)	Buena	Muy Baja (Desktop App)
SurfSense	Alta (Agente)	Excelente	Media

Mi recomendación final

Para tu caso concreto de **unificar perfiles y corregir disonancias**:

1. Prueba primero con Open Notebook: Al estar orientado a "cuadernos", te permite ir guardando tus hallazgos de similitudes y discrepancias como notas al margen mientras chateas con los documentos
2. Si necesitas algo más "automático", salta a Khoj. Su capacidad para crear agentes específicos te permitiría subir todos los Word/Excel y decirle: "Eres un consultor de organización. Analiza estos 50 perfiles y gener a una tabla comparativa de skills comunes

Un detalle importante para tus Excel: Las IAs suelen "sufrir" analizando archivos Excel crudos en modo RAG. Si tus documentos de perfiles tienen tablas complejas, te recomiendo convertirlos a formato **Markdown** ante s de subirlos a cualquiera de estas herramientas; esto mejora drásticamente la capacidad de la IA para entend er las jerarquías y relaciones entre celd
