

Introducción a Docker

UD 06. Caso práctico

08 - Ejecutando

DeepSeek R1 (Ollama) y

Open WebUI



Autor: Sergi García Barea

Actualizado Febrero 2025

Licencia



Reconocimiento – NoComercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

| | |
|---|---|
| 1. Introducción | 3 |
| 2. ¿Qué es Ollama? | 3 |
| 3. ¿Qué es DeepSeek R1? | 3 |
| 4. ¿Qué es Open WebUI? | 3 |
| 5. Fichero "docker-compose.yml" desplegar Ollama y Open WebUI | 3 |
| 6. Poniendo en marcha el sistema | 4 |
| 7. Descargando el modelo DeepSeek R1 | 4 |
| 8. Interactuando con el modelo | 5 |
| 9. Bibliografía | 5 |

UD06. CASO PRÁCTICO 08

1. INTRODUCCIÓN

En este caso práctico, aprenderemos a desplegar DeepSeek R1 (usando Ollama), un modelo de lenguaje de código abierto, junto con Open WebUI, una interfaz gráfica autoalojada y fácil de usar para interactuar con modelos LLM. Utilizaremos Docker Compose para facilitar la puesta en marcha y gestión de los contenedores.

2. ¿QUÉ ES OLLAMA?

Ollama es un gestor de modelos de lenguaje diseñado para ejecutar grandes modelos de inteligencia artificial en local. Facilita la descarga, ejecución y gestión de modelos sin necesidad de depender de servicios en la nube. Su arquitectura optimizada permite ejecutar modelos en hardware local con un alto nivel de eficiencia.

3. ¿QUÉ ES DEEPSEEK R1?

DeepSeek R1 es un modelo de inteligencia artificial desarrollado como alternativa de código abierto a los modelos comerciales de lenguaje natural. Se destaca por su accesibilidad y transparencia, permitiendo su uso en entornos de investigación y aplicaciones comerciales sin restricciones.

4. ¿QUÉ ES OPEN WEBUI?

Open WebUI es una interfaz gráfica de usuario diseñada para facilitar la interacción con modelos LLM sin necesidad de una conexión a la nube. Al ejecutarse localmente, permite mayor control sobre los datos y evita la dependencia de servicios externos.

Las ventajas de ejecutar estos modelos en local incluyen:

- **Personalización:** Permite ajustar modelos y configuraciones a necesidades específicas.
- **Costes reducidos:** Evita pagos por servicios en la nube si se dispone de una máquina adecuada.
- **Privacidad:** Garantiza que los datos no salgan del entorno local.

5. FICHERO "DOCKER-COMPOSE.YML" DESPLEGAR OLLAMA Y OPEN WEBUI

```
version: "3.9" # Versión de Docker Compose a utilizar

services:
  webui:
    image: ghcr.io/open-webui/open-webui:main # Imagen de Open WebUI desde GitHub Container Registry
    ports:
      - 3000:8080/tcp # Mapea el puerto 8080 del contenedor al puerto 3000 del host
    volumes:
      - open-webui:/app/backend/data # Volumen persistente para almacenar datos de Open WebUI
    extra_hosts:
      - "host.docker.internal:host-gateway" # Permite que el contenedor acceda al
```

```

host
  depends_on:
    - ollama # Indica que este servicio debe iniciarse después de 'ollama'

ollama:
  image: ollama/ollama # Imagen del servicio Ollama
  expose:
    - 11434/tcp # Expone el puerto 11434 dentro de la red interna de Docker
  ports:
    - 11434:11434/tcp # Mapea el puerto 11434 del contenedor al mismo puerto en el
host
  healthcheck:
    test: ollama --version || exit 1 # Verifica que Ollama esté funcionando
correctamente
  volumes:
    - ollama:/root/.ollama # Volumen persistente para almacenar datos de Ollama

volumes:
  ollama: # Define un volumen para Ollama, asegurando persistencia de datos
  open-webui: # Define un volumen para Open WebUI, evitando pérdida de información
entre reinicios

```

Este docker-compose.yml define dos servicios:

- **ollama**: Proporciona la infraestructura para gestionar y ejecutar modelos de lenguaje.
- **webui**: Ofrece una interfaz gráfica accesible en el puerto 3000 para interactuar con los modelos.

6. PONIENDO EN MARCHA EL SISTEMA

Para desplegar los servicios, situarnos en el directorio donde se encuentra el fichero docker-compose.yml y ejecutar:

```
docker compose up -d
```

Esto iniciará los contenedores de Ollama y Open WebUI. Una vez en marcha, accedemos a Open WebUI en <http://localhost:3000>.

7. DESCARGANDO EL MODELO DEEPSEEK R1

Al iniciar el sistema por primera vez, Open WebUI estará vacío ya que no incluye modelos por defecto. Aquí se pueden ver todos los modelos disponibles <https://ollama.com/library>

Para descargar DeepSeek R1, ejecutamos el siguiente comando dentro del contenedor de Ollama:

```
docker exec -it ollama-ollama-1 ollama pull deepseek-r1
```

El cual descargará el modelo por defecto de 7b de parámetros.

O si queremos solo probar el sistema y descargar una versión más ligera, podemos descargar la de 1.5b de parámetros

```
docker exec -it ollama-ollama-1 ollama pull deepseek-r1:1.5b
```


8. INTERACTUANDO CON EL MODELO

Podemos usar la interfaz gráfica en <http://localhost:3000> o interactuar directamente con el modelo desde la línea de comandos ejecutando:

```
docker exec -it ollama-ollama-1 ollama run deepseek-r1
```

o si procede, en su lugar

```
docker exec -it ollama-ollama-1 ollama run deepseek-r1:1.5b
```

 **Interesante:** Esta opción permite usar el modelo sin necesidad de una UI, lo que puede ser útil para automatizaciones o pruebas rápidas.

Finalmente, para detener y eliminar los contenedores sin afectar a los volúmenes de datos:

```
docker compose down
```

Esto mantendrá la persistencia de los datos en los volúmenes ollama y open-webui.

9. BIBLIOGRAFÍA

- [1] Docker Docs <https://docs.docker.com/>
- [2] Docker Compose Docs <https://docs.docker.com/compose/>
- [3] Open WebUI: <https://github.com/open-webui/open-webui>
- [4] DeepSeek: <https://www.deepseek.com/>
- [5] Ollama: <https://ollama.com/> y su lista de modelos <https://ollama.com/library>