

## 11 [IA] El cerebro de la bestia. IA en local

Comprobación previa a la instalación del sistema (obligatoria)

Verificar la versión del sistema

- lsb\_release -a

```
ub@ub:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04.3 LTS
Release:        24.04
Codename:       noble
```

Comprueba la arquitectura de la CPU

- uname -m

```
ub@ub:~$ uname -m
x86_64
```

Debe ser: x86\_64

Comprueba la memoria

free -h

```
ub@ub:~$ free -h
              total        usado        libre  compartido   búf/caché  disponible
Mem:          15Gi       1,5Gi       13Gi      40Mi       1,2Gi       14Gi
Inter:        3,6Gi        0B        3,6Gi
```

≥ 8 GB: capaz de ejecutar modelos pequeños

≥ 16 GB: capaz de ejecutar modelos como Llama 3 y Mistral

Instalación de Ollama en Ubuntu

- curl -fsSL https://ollama.com/install.sh | sh

```
ub@ub:~$ curl -fsSL https://ollama.com/install.sh | sh
>>> Installing ollama to /usr/local
[sudo] contraseña para ub:
>>> Downloading ollama-linux-amd64.tgz
##### 100.0%
```

Verifique si el servicio Ollama funciona correctamente

- systemctl status ollama

```
ub@ub:~$ systemctl status ollama
● ollama.service - Ollama Service
  Loaded: loaded (/etc/systemd/system/ollama.service; enabled; preset: enabled)
  Active: active (running) since Sun 2026-01-11 20:13:05 CET; 3min 10s ago
    Main PID: 4642 (ollama)
       Tasks: 9 (limit: 19036)
      Memory: 9.8M (peak: 20.9M)
         CPU: 71ms
```

## Ver versión

- ollama --version

```
ub@ub:~$ ollama --version
ollama version is 0.13.5
```

## Ejecutar modelos ligeros

- ollama run phi

```
ub@ub:~$ ollama run phi
pulling manifest
pulling 04778965089b: 100% [██████████] 1.6 GB
pulling 7908abcb772: 100% [██████████] 1.0 KB
pulling 774a15e6f1e5: 100% [██████████] 77 B
pulling 3188becd6bae: 100% [██████████] 132 B
pulling 0b8127ddf5ee: 100% [██████████] 42 B
pulling 4ce4b16d33a3: 100% [██████████] 555 B
verifying sha256 digest
writing manifest
success
>>> Send a message (/? for help)
```

## Ver modelos instalados

- ollama list

```
ub@ub:~$ ollama list
NAME          ID           SIZE      MODIFIED
phi:latest    e2fd6321a5fe  1.6 GB   11 minutes ago
```

## Descargar el modelo sin ejecutarlo

- ollama pull mistral

```
ub@ub:~$ ollama pull mistral
```

- ollama pull llama3:8b

```
ub@ub:~$ ollama pull llama3:8b
```

## Comprobación de la disponibilidad de la API local

- curl http://localhost:11434/api/tags

```
ub@ub:~$ curl http://localhost:11434/api/tags
[{"models": [{"name": "phi:latest", "model": "phi:latest", "modified_at": "2026-01-11T20:31:02.361279579+01:00", "size": 1602463378, "digest": "e2fd6321a5fe6bb3ac8a4e6f1cf04477fd2dea2924cf53237a995387e152ee9c", "details": {"parent_model": "", "format": "gguf", "family": "phi2", "families": ["phi2"], "parameter_size": "3B", "quantization_level": "Q4_0"}}]}]
```

## Editar servicios systemd

```
- sudo systemctl edit ollama
```

```
ub@ub:~$ sudo systemctl edit ollama
```

Añadir el siguiente contenido

```
[Service]
```

```
Environment="OLLAMA_HOST=0.0.0.0:11434"
```

```
# WantedBy=multi-user.target
```

```
[Service]
```

```
Environment="OLLAMA_HOST=0.0.0.0:11434"
```

Recargar y reiniciar

- sudo systemctl daemon-reload
- sudo systemctl restart ollama

```
[sudo] password for ub:
```

```
ub@ub:~$ sudo systemctl daemon-reload
```

```
ub@ub:~$ sudo systemctl restart ollama
```

Habilite el firewall y abra los puertos

- sudo ufw allow 11434/tcp

```
ub@ub:~$ sudo ufw allow 11434/tcp
```

```
Regla añadida
```

```
Regla añadida (v6)
```

Establecer la ruta de almacenamiento del modelo

- export OLLAMA\_MODELS=/mnt/data/ollama-models

```
ub@ub:~$ export OLLAMA_MODELS=/mnt/data/ollama-models
```

Instalar el modelo codellama

- ollama pull codellama

```
ub@ub:~$ ollama pull codellama
```

Uso directo (una sola petición)

- ollama run codellama "escribe un script en bash para hacer backup de /etc"

```
ub@ub:~$ ollama run codellama "escribe un script en bash para hacer backup de /etc"

Este es un ejemplo de cómo podrías crear un script en Bash para realizar una copia de seguridad de la carpeta '/etc':
```bash
#!/bin/bash

# Definir el directorio destino del backup
dest="/backup"

# Crear el directorio destino si no existe
if [ ! -d $dest ]; then
    mkdir $dest
fi

# Copiar la carpeta /etc al directorio destino
cp -r /etc/ $dest/etc
```

Este script utiliza el comando `mkdir` para crear un directorio llamado `/backup` si no existe, y luego utiliza el comando `cp` para copiar la carpeta `/etc` al directorio `/backup`. El parámetro `-r` se utiliza para indicar que queremos hacer una copia recursiva de la carpeta `/etc`, es decir, también queremos copiar todos los archivos y subcarpetas dentro de ella.

Ten en cuenta que este script solo crea un backup de la carpeta `/etc` y no realiza ninguna otra tarea de configuración o mantenimiento. Si deseas realizar una copia de seguridad de otras carpetas o hacer otras tareas, debes agregarlas al script según sea necesario.
```

Perfecto para redirecciones:

- ollama run codellama "crea un script de backup de /etc" > backup.sh
- chmod +x backup.sh

```
ub@ub:~$ ollama run codellama "crea un script de backup de /etc" > backup.sh

ub@ub:~$ 
ub@ub:~$ chmod +x backup.sh
```

Uso de Ollama en ejecución del sistema

- ollama run llama3 "Escribe un README.md sobre este proyecto" > README.md

```
ub@ub:~$ ollama run llama3 "Escribe un README.md sobre este proyecto" > README.md
```

- sudo nano ~/ollama-shell.sh

```
GNU nano 7.2                               /home/ub/ollama-shell.sh
#!/usr/bin/env bash

read -p "Pregunta a la IA: " pregunta
`respuesta=$(ollama run phi "$pregunta. Devuelve SÓLO un comando Linux seguro.")` 

echo
echo "La IA propone:"
echo "$respuesta"
echo

read -p "¿Ejecutar este comando? (s/n): " ok
if [[ "$ok" == "s" ]]; then
    bash -lc "$respuesta"
else
    echo "Cancelado."
fi
```

## Ver PATH

- echo \$PATH

```
ub@ub:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin  
:/snap/bin
```

Instala y utiliza el script Ollama Wrapper.

Mueve el script al directorio de comandos del sistema

- sudo mv ollama-shell.sh /usr/local/bin/ollama-shell

```
ub@ub:~$ sudo mv ollama-shell.sh /usr/local/bin/ollama-shell
```

Conceder privilegios de ejecución

- sudo chmod +x /usr/local/bin/ollama-shell

```
ub@ub:~$ sudo chmod +x /usr/local/bin/ollama-shell
```

Prueba en cualquier directorio

- cd /tmp
- ollama-shell

```
ub@ub:~$ cd /tmp  
ub@ub:/tmp$ ollama-shell  
Pregunta a la IA:
```

Script Wraper avanzado

```
GNU nano 7.2                               /home/ub/ollama-shell  
#!/usr/bin/env bash  
set -euo pipefail  
  
# -----  
# Ollama Shell Assistant (Seguro)  
# - Pregunta a un modelo Ollama local  
# - Fuerza español usando modelo "llama3-es"  
# - Devuelve un único comando  
# - Valida contra lista blanca  
# - Pide confirmación antes de ejecutar  
# - Registra todo en /var/log/ollama-shell.log  
# -----  
  
MODEL="${MODEL:-llama3-es}"  
LOG_FILE="${LOG_FILE:-/var/log/ollama-shell.log}"  
  
MODE="${MODE:-ask}" # ask | exec  
TIMEOUT_SECS="${TIMEOUT_SECS:-120}"  
  
# Lista blanca básica de comandos permitidos (ajústala a tu contexto)
```

Conceder privilegios de ejecución

```
ub@ub:~$ sudo chmod +x ~/ollama-shell
```

Mover a /usr/local/bin

- sudo mv ~/ollama-shell /usr/local/bin/ollama-shell

```
ub@ub:~$ sudo mv ~/ollama-shell /usr/local/bin/ollama-shell
```

Volver a conceder permisos

- sudo chmod +x /usr/local/bin/ollama-shell

```
ub@ub:~$ sudo chmod +x /usr/local/bin/ollama-shell
```

Prueba:

- ollama-shell "muestra los puertos en escucha"

```
ub@ub:/tmp$ ollama-shell "muestra los puertos en escucha"
```

Para permitir ejecución (con confirmación):

- ollama-shell --exec "muestra los últimos intentos de login fallidos"

```
ub@ub:~$ ollama-shell --exec "muestra los últimos intentos de login fallidos"
```

Forma más simple: usar el modelo por defecto

- ollama-shell "lista los usuarios conectados"

```
ub@ub:~$ ollama-shell "lista los usuarios conectados"
```

Indicar el modelo en la llamada (recomendado)

- ollama-shell -m codellama-es "crea un script bash para hacer backup de /etc"

```
ub@ub:~$ ollama-shell -m codellama-es "crea un script bash para hacer backup de /etc"
```

Forzar el modelo con variable de entorno

- MODEL=codellama-es ollama-shell "analiza este script"

```
ub@ub:~$ MODEL=codellama-es ollama-shell "analiza este script"
```

Para toda la sesión

- export MODEL=codellama-es
- ollama-shell "genera un Makefile sencillo"

```
ub@ub:~$ export MODEL=codellama-es
```

```
ub@ub:~$ ollama-shell "genera un Makefile sencillo"
```

Cambiar el modelo por defecto en el script

- sudo nano /usr/local/bin/ollama-shell

```
ub@ub:~$ sudo nano /usr/local/bin/ollama-shell
```

Busca:

MODEL="\${MODEL:-llama3-es}"

Y cambia a:

MODEL="\${MODEL:-codellama-es}"

```
GNU nano 7.2                               /usr/local/bin/ollama-shell *
#!/usr/bin/env bash
set -euo pipefail

# -----
# Ollama Shell Assistant (Seguro)
# - Pregunta a un modelo Ollama local
# - Fuerza español usando modelo "llama3-es"
# - Devuelve un único comando
# - Valida contra lista blanca
# - Pide confirmación antes de ejecutar
# - Registra todo en /var/log/ollama-shell.log
# -----


MODEL="${MODEL:-codellama-es}" ←
LOG_FILE="${LOG_FILE:-/var/log/ollama-shell.log}"

MODE="${MODE:-ask}" # ask | exec
TIMEOUT_SECS="${TIMEOUT_SECS:-120}"
```

Ver qué modelo se está usando

- echo "Modelo en uso: \$MODEL"

```
ub@ub:~$ echo "Modelo en uso: $MODEL"
Modelo en uso: codellama-es
ub@ub:~$
```