

## Condizioni d'uso dell'Assistente IA (Antigravity/Gemini)

Poiché l'elaborazione dei modelli su cui mi baso (Gemini di Google) avviene principalmente in cloud (sui server di Google), non presento rigidi requisiti hardware (CPU, RAM o GPU) a differenza dei modelli eseguiti in locale. Le condizioni d'uso dipendono per lo più dall'applicazione o interfaccia client utilizzata per interagire con me (es. browser web, estensioni per IDE come Visual Studio Code).

Di seguito le condizioni per ogni sistema operativo.

### Windows

- **Versioni OS supportate:** Windows 10 (aggiornato a versioni recenti, preferibilmente 64-bit) o Windows 11. Le versioni più vecchie come Windows 7 e 8 non garantiscono più il supporto per i browser o gli IDE aggiornati.
- **Browser web:** Ultime versioni di Google Chrome, Microsoft Edge o Mozilla Firefox.
- **IDE (se mi usi come plugin per sviluppatori):** Visual Studio Code, Cursor, o altri IDE supportati aggiornati ad una release recente (generalmente l'ultimo anno).
- **Connettività:** Indispensabile una connessione Internet stabile.

### macOS

- **Versioni OS supportate:** macOS 11 (Big Sur) o versioni successive. Un sistema operativo aggiornato è fondamentale per supportare gli aggiornamenti di sicurezza dei browser.
- **Architettura:** Piena compatibilità sia per i Mac basati su processori Intel, sia per i chip Apple Silicon (M1, M2, M3, M4).
- **Browser web:** Safari (aggiornato), Google Chrome, Microsoft Edge, o Mozilla Firefox.
- **Connettività:** Essenziale la connessione a Internet.

### Linux

- **Versioni OS supportate:** Distribuzioni moderne a 64-bit supportate dai principali browser e strumenti di sviluppo (es. Ubuntu 20.04 LTS o superiore, Debian 11+, Fedora 36+).

- **Gestione Finestratura:** Supportato sia su server grafici X11 che Wayland, dipendentemente dal browser.
- **Browser web:** Google Chrome/Chromium, Mozilla Firefox.
- **Connettività:** Essenziale la connessione a Internet.

[!TIP] **Riepilogo Generale:** Finché il sistema operativo è in grado di eseguire versioni moderne e sicure dei browser web o degli ambienti di sviluppo (IDE), sarai in grado di interagire e sfruttare al 100% l'intelligenza artificiale in cloud.

## Condizioni per utilizzare Ollama

Ollama è uno strumento che permette di eseguire modelli linguistici di grandi dimensioni (LLM) interamente **in locale** sul tuo dispositivo. Di conseguenza, a differenza delle IA in cloud, richiede risorse hardware specifiche, in particolar modo a livello di memoria RAM e possibilmente accelerazione GPU (scheda video).

### Requisiti Hardware Comuni (Tutti gli OS)

- **RAM di sistema:** Almeno 8 GB per eseguire modelli leggeri (fino a 7B parametri, es. Llama 3 8B, Mistral). Consigliati 16 GB per modelli da 13B parametri, e 32 GB per modelli da 33B.
  - **Spazio di archiviazione:** Essendo scaricati in locale, ogni modello occupa da 4 GB a decine di GB. È fortemente raccomandato un disco a stato solido (SSD).
- 

### Windows

- **Versioni del Sistema Operativo:** Windows 10 o Windows 11.
- **Integrazione:** Originariamente Ollama richiedeva WSL 2 (Windows Subsystem for Linux), ma ora esiste un **eseguibile nativo** per Windows da installare in maniera tradizionale.
- **Accelerazione Hardware:**
  - **NVIDIA:** Pieno supporto tramite CUDA (sono necessari i driver NVIDIA più recenti).
  - **AMD:** Supporto per schede AMD Radeon tramite la componente ROCm implementata in Ollama per Windows.
  - Se non si possiede una GPU supportata, Ollama funzionerà esclusivamente sulla CPU risultando sensibilmente più lento.

### macOS

- **Versioni del Sistema Operativo:** macOS 11 (Big Sur) o versioni più recenti.
- **Architettura:**

- **Apple Silicon (M1, M2, M3, M4, e varianti Pro/Max/Ultra):** Questa è la condizione ideale per eseguire Ollama. Grazie all'architettura a memoria unificata, Ollama sfrutta l'accelerazione GPU integrata di Apple con prestazioni eccezionali.
- **Mac con processori Intel:** Supportati, ma senza una GPU discreta supportata, i calcoli avverranno via CPU, il che farà aumentare molto i tempi di risposta.
- È un'applicazione nativa distribuita direttamente in pratico `.zip` o installer per macOS.

## Linux

- **Versioni del Sistema Operativo:** Generalmente qualsiasi moderna distribuzione Linux (es. Ubuntu, Debian, Fedora, Arch Linux).
- **Installazione:** Molto rapida, viene di solito eseguita tramite un semplice script da terminale: `curl -fSSL https://ollama.com/install.sh | sh`
- **Accelerazione Hardware:**
  - **NVIDIA:** È strettamente necessario avere il driver NVIDIA installato sul proprio kernel e preferibilmente il NVIDIA Container Toolkit (se usato via Docker) oppure direttamente CUDA.
  - **AMD:** Supportato tramite drivers ROCm.
  - Il demone (`ollama.service`) eseguito su Linux è molto flessibile e permette configurazioni avanzate via `systemctl`.

[!NOTE] Su tutti i sistemi operativi, Ollama si interfaccia come un servizio server su `http://localhost:11434`, consentendo al terminale e alle applicazioni terze di collegarsi tramite le sue API Rest.

## Autorizzare l'Accesso dei Browser e app a Ollama (CORS e Rete)

Spesso, quando un browser web o un'applicazione cerca di collegarsi a Ollama (11434), la richiesta viene bloccata. I problemi principali sono due:

1. **Errore CORS (Cross-Origin):** il browser blocca la richiesta per sicurezza se non esplicitamente autorizzato. Variabile responsabile: `OLLAMA_ORIGINS`.

2. **Errore di Rete / Connessione Rifiutata (Localhost):** se provi a raggiungere Ollama da un PC/Smartphone diverso o da Docker. Variabile responsabile: OLLAMA\_HOST.

Ecco come configurare entrambe le protezioni per permettere al browser di accedere senza restrizioni.

---

## Windows

1. Premi Start e scrivi **Variabili d'ambiente**. Poi clicca su "Modifica le variabili dell'ambiente di sistema".
2. Clicca sul pulsante **Variabili d'ambiente...**
3. In una delle sezioni ("Variabili dell'utente" o "di sistema") aggiungi due nuove variabili cliccando su **Nuova...**:

### **Per rimuovere il blocco ai browser (CORS e file locali/HTML):**

- Nome variabile: OLLAMA\_ORIGINS
- Valore variabile: \* (*Nota: scrivi solo l'asterisco, senza virgolette*)

### **Se hai bisogno di accedere a Ollama da un altro PC nella stessa rete Wi-Fi/LAN (Opzionale):**

- Nome variabile: OLLAMA\_HOST
- Valore variabile: 0 . 0 . 0 . 0

4. Clicca OK e chiudi le finestre.
  5. **RIAVVIO FONDAMENTALE:** Non basta "chiudere" normalmente Ollama.
    - Vai in basso a destra (accanto all'orologio), fai clic destro sull'icona di Ollama e clicca su **Quit Ollama**.
    - **Consigliato:** Se dopo averlo riaperto non funziona ancora, *riavvia direttamente l'intero computer*, in quanto Windows a volte non aggiorna le variabili di sistema fino al riavvio.
-

## macOS (Terminale)

1. Apri il **Terminale** e digita questi comandi (premi Invio dopo ciascuno):

```
launchctl setenv OLLAMA_ORIGINS "*"
launchctl setenv OLLAMA_HOST "0.0.0.0"
```

2. Chiudi l'app di Ollama: clicca sull'icona in alto a destra nella menu bar del Mac e seleziona **Quit Ollama**.
3. Riavvia Ollama. *(Nota: Se al riavvio del Mac il problema si ripresenta, le variabili potrebbero dover essere reinserite, o aggiunte permanentemente a uno script di avvio o nel .zshrc).*

---

## Linux (Servizio Systemd)

1. Apri il terminale per sovrascrivere la configurazione del servizio:

```
sudo systemctl edit ollama.service
```

2. Scrivi questo esatto blocco all'interno del file (sotto a [Service]), cancellando eventuali cancelletti iniziali se presenti:

```
[Service]
Environment="OLLAMA_ORIGINS=*"
Environment="OLLAMA_HOST=0.0.0.0"
```

3. Salva e chiudi l'editor (su Nano: premi CTRL+O, Invio, poi CTRL+X).
4. Ricarica e riavvia:

```
sudo systemctl daemon-reload
sudo systemctl restart ollama
```

---

[!WARNING] **Estensioni Chrome/Firefox:** Se usi estensioni, a volte l'asterisco \* non basta. In quel caso, invece di \*, prova a impostare il valore esatto dell'estensione, ad esempio: `OLLAMA_ORIGINS="chrome-extension://*, moz-extension://*" Pagine File (app .html locali): A volte i browser moderni trattano severamente il protocollo file:/// . Aprire un .html dal desktop invocando`

OLLAMA\_ORIGINS="" \* "" dovrebbe risolverlo, ma assicurati sempre di chiudere e riaprire davvero Ollama!

