Linux + IA 2025/05/27

Personalizando Modelfile Ollama

- Tomando como base el archivo descargado Llama-3.2-3B-Instruct-Q4_K_M.gguf
- Dentro de la carpeta gguf/
  - cd ~/gguf
  - crear el archivos Modelfile

    - Modelfile

```
FROM ./Llama-3.2-3B-Instruct-Q4_K_M.gguf


SYSTEM """
Eres un asistente técnico especializado exclusivamente en
Ubuntu 24.04 LTS.
Tu función es responder preguntas sobre instalación,
administración, troubleshooting, redes, systemd, paquetes
APT, permisos, usuarios, servicios, Docker, Podman y
seguridad básica en Ubuntu 24.04.

Reglas:
- Responde siempre en español.
- Si la pregunta no está relacionada con Ubuntu 24.04,
indica que solo respondes sobre Ubuntu 24.04.
- Da respuestas prácticas, directas y con pasos
concretos.
- Cuando sea útil, muestra comandos listos para copiar y
pegar.
- Si un comando puede ser destructivo, advierte antes de
usarlo.
- Si faltan datos, haz una sola pregunta de aclaración.
- Actúa como agente técnico: propone diagnóstico,
verificación y corrección.
- Prioriza soluciones seguras y compatibles con Ubuntu
24.04 LTS.
"""

TEMPLATE """
{{ if .System
}}<|start_header_id|>system<|end_header_id|>
{{ .System }}<|eot_id|>{{ end }}
{{ if .Prompt }}<|start_header_id|>user<|end_header_id|>
{{ .Prompt }}<|eot_id|>{{ end }}
<|start_header_id|>assistant<|end_header_id|>
"""
```

```
PARAMETER temperature 0.2
PARAMETER top_p 0.9
PARAMETER num_ctx 4096
PARAMETER repeat_penalty 1.1
PARAMETER num_predict 512
```

- Generar el nuevo modelo
  - `ollama create llamaubuntu24 -f Modelfile`
- Listamos
  - `ollama list`
- Utilizamos
  - `ollama run llamaubuntu24:latest`
- Detener el modelo para despues ejecutarlo con llama.cpp
  - `ollama stop llamaubuntu24:latest`

Personalizando llama.cpp

- No se puede utilizar un GGUF modificado ya que no toma en cuenta las modificaciones.
- La forma correcta es pasarle en un prompt inicial las limitaciones
  - `llama-cli -m Llama-3.2-3B-Instruct-Q4_K_M.gguf -p "Eres un asistente técnico especializado exclusivamente en Ubuntu 24.04 LTS." -n 512 -cnv`

* Para que sea persistente se deben realizar las siguientes modificaciones.

- En el mismo lugar del script start-llama.sh crear el archivo
  - nano template.json
  - template.json

```
{
  "system_message": "Eres un asistente técnico especializado exclusivamente en Ubuntu 24.04 LTS. Da respuestas prácticas, directas y con pasos concretos. Sin explicaciones. Si un comando puede ser destructivo, advierte antes de usarlo.
  "
}
```

- Copiar el archivo

    - cp start-llama.sh start-llama-ubuntu.sh
- Editarlo
    - start-llama-ubuntu.sh

```
#!/bin/bash

# Cambiar esta ruta cuando quieras otro modelo
MODEL="$HOME/gguf/Llama-3.2-3B-Instruct-Q4_K_M.gguf"

nohup llama-server \
    --jinja \
    -m "$MODEL" \
    --chat-template "custom" \
    --chat-template-file ./template.json \
    --host 127.0.0.1 \
    --port 11435 \
    -c 8192 \
    -t 4 \
    > /tmp/llama-server.log 2>&1 &

echo $! > /tmp/llama-server.pid
echo "Servidor iniciado. PID: $(cat /tmp/llama-server.pid)"
```

Nvidia llama.cpp

- Requerimiento previo:
    - Tener configurado correctamente los drivers de Nvidia.
    - Y tener actualizado el sistema.
    - Verificación ejecutar

```
nvidia-smi
```

    -
```
sudo apt install -y nvidia-cuda-toolkit
nvcc --version
```

- Compilación
    -
```
git clone https://github.com/ggerganov/llama.cpp
cd llama.cpp
cmake -B build -DGGML_CUDA=ON -DCMAKE_CUDA_ARCHITECTURES="86" -
DCMAKE_BUILD_TYPE=Release
cmake --build build --config Release -j$(nproc)
```

- Adicionar PATH
    -
```
nano .bashrc
```

- Adicionar al final

```
export PATH="$HOME/llama.cpp/build/bin:$PATH"
```

-
```
source .bashrc
```

- Primera consulta
  -
```
llama-cli --version
llama-cli -m gguf/Llama-3.2-3B-Instruct-Q4_K_M.gguf --gpu-layers
-1 -p "Hola" -n 50
```