Nos centraremos en modelos de lenguaje grandes (LLMs) que pueden correr en CPUs modestas.

Manual Básico: Tu Propia IA Conversacional Local

Objetivo: Aprender a descargar, instalar, ejecutar y personalizar un modelo de Inteligencia Artificial en tu propio ordenador.

¿Qué necesitas antes de empezar? (Requisitos Previos)

1. **Un Ordenador:**
    - **Sistema Operativo:** Windows, macOS (con chip M es ideal) o Linux.
    - **CPU:** Con al menos 4 hilos (la mayoría de los procesadores de los últimos 8-10 años cumplen).
    - **RAM:** Mínimo 8 GB. 16 GB es recomendable para modelos un poco más grandes o para mayor fluidez.
    - **Disco Duro:** Al menos 20-30 GB de espacio libre (los modelos pueden ocupar varios GB).
    - **Conexión a Internet:** Para descargar el software y los modelos.
2. **Software Básico (si no lo tienes ya):**
    - Un navegador web (Chrome, Firefox, Edge, etc.).
    - (Opcional, pero muy recomendado para algunas partes) Git: https://git-scm.com/downloads
    - (Opcional, para la parte de entrenamiento con text-generation-webui) Python: https://www.python.org/downloads/ (elige la versión estable más reciente).

Parte 1: Descargar y Ejecutar tu Primera IA (Chat de Texto)

Vamos a usar herramientas que facilitan este proceso. Tienes dos opciones principales para empezar:

Opción A: LM Studio (Recomendado para principiantes por su interfaz gráfica)

LM Studio es una aplicación que te permite descubrir, descargar y ejecutar LLMs fácilmente.

1. **Descargar e Instalar LM Studio:**

- Ve a https://lmstudio.ai/.
- Descarga la versión para tu sistema operativo (Windows, macOS, Linux).
- Instala la aplicación como cualquier otro programa.

2. **Buscar y Descargar un Modelo:**
   - Abre LM Studio.
   - En la pantalla principal (o la pestaña con el icono de lupa □), busca un modelo. Para empezar, te recomiendo modelos pequeños pero capaces:
     - Phi-3-Mini (busca por ejemplo "Phi-3 Mini Instruct GGUF")
     - TinyLlama (busca por ejemplo "TinyLlama Chat GGUF")
   - **Importante: Formato GGUF.** Asegúrate de que los archivos del modelo que elijas estén en formato GGUF. Este formato está optimizado para correr en CPU.
   - **Importante: Cuantización.** Verás versiones como Q4_K_M, Q5_K_M, Q3_K_S. Estos indican el nivel de compresión (cuantización). Q4_K_M es un buen equilibrio entre rendimiento y tamaño para CPUs. Un modelo Phi-3 Mini Q4_K_M ocupará unos ~2.2GB.
   - Haz clic en "Download" en la versión del modelo que elijas. Espera a que se complete la descarga.
3. **Chatear con tu IA:**
   - Ve a la pestaña de Chat (icono de bocadillo □ a la izquierda).
   - Arriba, en el centro, selecciona el modelo que acabas de descargar.
   - ¡Listo! Escribe en el cuadro de texto de abajo y presiona Enter para conversar con tu IA local.
4. **(Opcional) Iniciar el Servidor Local en LM Studio:**
   - Para que otras herramientas (como las de voz o entrenamiento) puedan usar este modelo, ve a la pestaña del Servidor Local (icono <> a la izquierda).
   - Selecciona tu modelo.
   - Haz clic en "Start Server". Esto hace que tu modelo esté disponible en una dirección como http://localhost:1234. Anota esta dirección.

Opción B: Ollama (Para quienes se sienten cómodos con la línea de comandos)

Ollama es una herramienta de línea de comandos muy popular y eficiente.

1. **Descargar e Instalar Ollama:**
    - Ve a https://ollama.com/.
    - Sigue las instrucciones de descarga e instalación para tu sistema operativo.
2. **Descargar un Modelo:**
    - Abre una terminal (Símbolo del sistema o PowerShell en Windows, Terminal en macOS/Linux).
    - Escribe el comando para descargar un modelo. Ejemplos:
        - Para Phi-3 Mini (versión optimizada por defecto): ollama pull phi3:mini
        - Para TinyLlama: ollama pull tinyllama
    - Ollama buscará y descargará la versión GGUF adecuada.
3. **Chatear con tu IA:**
    - En la misma terminal, escribe:
        - ollama run phi3:mini (o el nombre del modelo que descargaste).
    - Escribe tu mensaje y presiona Enter. Para salir del chat, escribe /bye.

Parte 2: "Entrenar" tu IA Local (Personalización)

"Entrenar" puede significar dos cosas principales en este contexto:

1. **Fine-tuning (Ajuste Fino):** Modificar ligeramente los "pesos" internos del modelo para que se adapte mejor a un estilo o conjunto de datos específico. Usaremos **LoRA (Low-Rank Adaptation)**, que es una técnica eficiente para esto.
2. **RAG (Retrieval Augmented Generation):** Darle al modelo acceso a tus propios documentos para que pueda responder preguntas basándose en ellos, sin modificar el modelo en sí.

Opción A: Fine-tuning Ligero con LoRA (usando text-generation-webui)

text-generation-webui (también conocido como Oobabooga) es una interfaz web muy completa para interactuar y entrenar LLMs.

1. **Instalar text-generation-webui:**

- Abre una terminal.
- Clona el repositorio (necesitas Git instalado): git clone https://github.com/oobabooga/text-generation-webui.git
- Entra en la carpeta: cd text-generation-webui
- Sigue las instrucciones de instalación de su README. Usualmente implica ejecutar un script como start_windows.bat, start_macos.sh o start_linux.sh. Esto puede tardar un rato ya que descargará dependencias (necesitas Python).
- Una vez instalado, ejecuta el mismo script para iniciar la interfaz web. Se abrirá en tu navegador en una dirección como http://127.0.0.1:7860.

2. **Preparar tus Datos de Entrenamiento:**
    - Para LoRA, necesitas datos en un formato simple, como un archivo CSV con dos columnas: instruction (la pregunta o instrucción) y response (la respuesta deseada).
    - Ejemplo (mis_datos.csv):

Fragmento de código

instruction,response

"¿Cuál es el lema de nuestro equipo?","¡Guerreros de la IA, forjando el futuro!"

"Describe nuestro proyecto principal.","Estamos construyendo un asistente local de IA con voz."

    - Cuantos más ejemplos de calidad tengas, mejor (decenas o cientos es un buen comienzo para LoRA).

3. **Cargar tu Modelo Base en text-generation-webui:**
    - Ve a la pestaña "Model".
    - Selecciona el tipo de modelo llama.cpp (para GGUF).
    - Busca y selecciona el archivo GGUF de tu modelo (el que descargaste con LM Studio o el que usa Ollama – los de Ollama suelen estar en ~/.ollama/models/manifests/... y luego en ~/.ollama/models/blobs/..., aunque puede ser más fácil usar el descargado con LM Studio).
    - Ajusta parámetros como n_gpu_layers a 0 si solo usas CPU. n_ctx es el tamaño del contexto (ej. 2048 o 4096).
    - Haz clic en "Load".

4. **Entrenar un LoRA:**
    - Ve a la pestaña "Training", luego a la sub-pestaña "LoRA".
    - Sube tu archivo CSV de datos.
    - Configura los parámetros. Para empezar:
        - **LoRA name:** Dale un nombre a tu LoRA (ej. mi_IA_personalizada).
        - **Rank (Dimension):** Un valor como 8, 16 o 32.
        - **Alpha:** Usualmente el mismo valor que Rank.
        - **Epochs:** Cuántas veces recorrer tus datos (empieza con 3-5).
        - **Learning Rate:** Algo como 1e-4 o 2e-4.
    - Haz clic en "Start LoRA Training". Esto puede tardar dependiendo de tus datos y CPU.
    - El LoRA entrenado se guardará (generalmente en la carpeta loras dentro de text-generation-webui).
5. **Usar tu LoRA Entrenado:**
    - En la pestaña "Model", después de cargar tu modelo base, busca la sección "LoRA" a la derecha.
    - Selecciona tu LoRA de la lista desplegable y haz clic en "Apply LoRAs".
    - Ahora, cuando chatees (en la pestaña "Text generation"), el modelo debería reflejar la personalización de tu LoRA.

Opción B: Añadir Conocimiento con RAG (usando Flowise o Langflow)

Esta opción no "entrena" el modelo en sí, sino que le da acceso a tus documentos. Es ideal si quieres que responda preguntas sobre PDFs, webs, etc.

1. **Instalar Flowise o Langflow:**
    - Son herramientas "low-code" para construir flujos de IA.
    - **Flowise (Node.js):**
        - Necesitas Node.js y npm.
        - En terminal: npm install -g flowise
        - Luego: npx flowise start (se abre en http://localhost:3000)

- **Langflow (Python):**
    - Necesitas Python y pip.
    - En terminal: pip install langflow
    - Luego: langflow (se abre en http://localhost:7860 u otro puerto)

2. **Construir un Flujo RAG:**
    - Dentro de Flowise/Langflow:

1. **Cargador de Documentos:** Añade nodos para cargar tus PDFs, TXTs, o URLs.

2. **Divisor de Texto:** Para fragmentar documentos grandes.

3. **Embeddings:** Un modelo que convierte texto en vectores numéricos (puedes usar modelos de embedding locales o apuntar a tu servidor de LM Studio/Ollama si tiene un endpoint de embeddings).

4. **Vector Store:** Una base de datos para estos vectores (ej. Chroma, FAISS, In-memory).

5. **LLM:** Conecta aquí tu modelo LLM local (apuntando al servidor de LM Studio o al API de Ollama, que es un poco más avanzado de configurar).

6. **Cadena RAG/Retriever:** Un nodo que toma la pregunta del usuario, busca información relevante en el Vector Store y la pasa al LLM junto con la pregunta.

    - Consulta tutoriales específicos de Flowise o Langflow para RAG, ya que la interfaz y los nodos exactos varían.

Parte 3: (Avanzado) Hacer que tu IA "Hable" con Voz

Una vez que tienes tu IA respondiendo por texto, puedes añadirle voz. Esto implica dos componentes:

1. **Speech-to-Text (STT):** Convertir tu voz en texto.
    - **Herramienta:** Whisper.cpp es excelente, ligera y funciona offline.
    - **Proceso:** Compilar Whisper.cpp, descargar un modelo de voz (ej. ggml-base.en.bin), y usarlo para transcribir audio de tu micrófono o un archivo.
2. **Text-to-Speech (TTS):** Convertir el texto de la IA en voz.
    - **Herramienta:** Piper es muy buena, ofrece voces naturales y es eficiente.

    - **Proceso:** Ejecutar Piper (fácil con Docker), enviarle texto, y te devolverá un archivo de audio con la voz.

Integración: Necesitarás un script o una herramienta (como Flowise con nodos específicos, si los tiene) que:

1. Capture audio de tu micrófono.
2. Lo envíe a Whisper.cpp (STT) para obtener texto.
3. Envíe ese texto a tu LLM local (LM Studio/Ollama).
4. Tome la respuesta de texto del LLM.
5. La envíe a Piper (TTS) para generar audio.
6. Reproduzca ese audio.

Consejos Finales:

- **Paciencia:** Trabajar con IA local implica aprender y a veces solucionar problemas. ¡No te desanimes!
- **Empieza Simple:** Con LM Studio y un modelo pequeño. Luego avanza gradualmente.
- **Documentación:** Cada herramienta tiene su propia documentación. Consulta sus páginas de GitHub o sitios web.
- **Comunidad:** Busca foros o comunidades online (Reddit, Discord) sobre estas herramientas. Mucha gente comparte sus experiencias y soluciones.
- **Hardware:** Si notas que va muy lento, puede ser por las limitaciones de tu hardware. Modelos más pequeños y cuantizaciones más agresivas (ej. Q3_K_S en GGUF) pueden ayudar, a costa de algo de calidad.

¡Mucha suerte en tu aventura de forjar tu propia IA local! Es un campo fascinante y muy empoderador.