

Aplicación de la IA en el aula: Cientificotecnológico

Tercer módulo perteneciente al itinerario formativo en alfabetización de la IA en el ámbito educativo, específico de cada perfil docente.

- [1. Panorámica tecnologías de IA](#)
 - [Creación de recursos](#)
 - [Modelos de IA](#)
 - [Plataformas y herramientas](#)
 - [Técnicas de IA avanzados](#)
 - [Infraestructura de IA](#)
 - [Fuentes de datos](#)
 - [Tablas comparativas](#)

- [2. Fundamentos de Machine Learning y Deep Learning](#)
 - [Aprender para y sobre la IA](#)

- [4. Prompting, RAG, Fine Tuning](#)
 - [Herramientas y plataformas](#)

- [5. Uso de agentes y automatización](#)
 - [Actividad 1](#)

- [6. Aplicaciones de IA en el ámbito científico](#)
 - [Casos de uso Generales](#)
 - [Casos de uso en Biología y Geología](#)
 - [Casos de uso en Física y Química](#)
 - [Casos de uso en Matemáticas](#)

1. Panorámica tecnologías de IA

1. Panorámica tecnologías de IA

Creación de recursos

Resumen ejecutivo

- **Modelos de IA:** Incluyen grandes modelos lingüísticos (LLM) para texto (GPT, BERT, etc.), modelos de visión (CNN, Diffusion) y modelos multimodales que combinan texto e imagen. Los *modelos fundacionales* son gigantescas redes pre-entrenadas en datos masivos que luego se adaptan a tareas específicas (mediante fine-tuning).
- **Plataformas:** Hay entornos colaborativos en la nube y herramientas locales. Ejemplos destacados son Hugging Face (repositorio de 2M modelos, demos y Spaces de IA), LangChain (framework de encadenado de LLMs con fuentes de datos), Google Colab y Kaggle Notebooks (Jupyter en la nube con GPU/TPU gratuitas), servicios ML empresariales como AWS SageMaker o Azure ML (notebooks integrados y despliegue en producción), y aplicaciones de IA local como Ollama o LM Studio (herramientas gratuitas para correr LLMs open-source en PCs). La librería **vLLM** destaca como motor optimizado para servir LLMs con alta eficiencia de memoria y rendimiento.
- **Librerías:** Las más usadas incluyen **PyTorch** (“biblioteca optimizada de tensores para DL en GPU/CPU”), **TensorFlow** (plataforma de Google para crear modelos ML ejecutables en cualquier entorno) y **Scikit-learn** (biblioteca Python de ML tradicional para regresión, clasificación, clustering, etc. de código abierto).
- **Técnicas clave:** RAG (Generación Aumentada con Recuperación) combina LLMs con búsquedas en bases de datos externas para anclar las respuestas en información actualizada. El *fine-tuning* es reentrenar parcialmente un modelo pre-entrenado en datos específicos. *Prompt engineering* consiste en diseñar con cuidado las instrucciones al modelo. Métodos recientes como **LoRA** (Low-Rank Adaptation) añaden matrices de bajo rango a un LLM para adaptarlo con muy pocos parámetros (ganando eficiencia de memoria). En general, las aproximaciones **PEFT** (“Parameter-Efficient Fine-Tuning”) ajustan solo unos pocos parámetros extra del modelo, congelando el resto, lo que reduce drásticamente los costos de cómputo y almacenamiento.
- **Infraestructura y hardware:** Hay aceleradores especializados para IA. **GPUs** (p.ej. NVIDIA H100/A100) son procesadores paralelos versátiles con gran memoria (80–141 GB) y fuerte ecosistema (CUDA, PyTorch). **TPUs** (Google TPU v6e, etc.) son ASICs de tensor optimizados para cargas de inferencia en nube: p.ej. v6e ofrece ~2 PFLOPS FP16 por ~\$2.70/h, pero solo están en Google Cloud (Google Drive, Colab). **NPU**s (Unidades de Procesamiento Neural) integradas en SoCs de móviles/PCs (por ejemplo, el *Apple Neural*

Engine en iPhones/Mac) ofrecen muy alta eficiencia energética: la 5ª generación de Apple NPU alcanza ~15.8 TFLOPS FP16. **FPGAs** y **ASICs** (como Google Edge TPU) permiten IA embebida con baja latencia, aunque son menos flexibles y a menudo caros. Los aceleradores neuromórficos (Intel Loihi, IBM TrueNorth) son experimentales para IA inspirada en el cerebro. Cada hardware tiene su costo/rendimiento: las GPUs de consumo cuestan cientos de euros, TPUs cobran por hora en nube, NPUs vienen “incluidas” en dispositivos móviles, FPGAs/ASICs son especializados (educativamente poco comunes), etc.

- **Ejecución de modelos:** Se puede en **la nube**, **en local** o en **el edge**. La nube (Colab, AWS, Azure) ofrece escalabilidad y cero instalación, pero implica costos y dependencia de Internet. En local (PC o servidores propios) hay más control de datos y cero latencia de red, pero requiere disponer de hardware adecuado. En el edge (móviles, IoT) se prioriza rapidez y privacidad (los datos no salen del dispositivo). Por ejemplo, la startup española Multiverse destaca que ejecutar IA en el dispositivo (“edge”) mejora la eficiencia energética y la privacidad.
- **Datos y conjuntos:** La calidad de los datos es crucial. Existen repositorios abiertos (Hugging Face Datasets, Kaggle Datasets, conjuntos académicos como ImageNet, UCI, NOAA, etc.) en múltiples disciplinas. Hay que asegurar la ética (consentimiento de datos, evitar sesgos, transparencia) y cumplir normativas de gobernanza (GDPR en Europa, futuras leyes de IA).

[image.png](#)

1. Panorámica tecnologías de IA

Modelos de IA

Los **LLMs (Large Language Models)** son modelos de lenguaje de gran tamaño (GPT, BERT, LLaMA, etc.) entrenados con enormes corpus de texto para tareas como generación de texto, traducción o respuesta a preguntas. Hugging Face describe la biblioteca *Transformers* que gestiona estos modelos, indicando que éstos se pueden usar en modalidades diversas: texto, imágenes, audio e incluso enfoques multimodales. Además de los LLMs “puro texto”, existen **modelos de visión** (CNNs, Vision Transformers, difusión estable) para clasificación/segmentación de imágenes, y **modelos multimodales** (p.ej. CLIP, LLaVA) que procesan simultáneamente texto e imagen. Todos ellos suelen derivar de un **modelo fundacional (foundation model)**: grandes redes preentrenadas con datos genéricos a escala (por ejemplo OpenAI GPT o Google BERT). Estos modelos genéricos proporcionan las capacidades de base, que luego se ajustan (fine-tuning) o especializan para tareas concretas. Por ejemplo, un “modelo fundacional” genérico entrenado con texto diverso se puede luego finetunear para responder solo preguntas médicas. En resumen, los modelos actuales de IA suelen ser *preentrenados* en datos amplios y luego usados o adaptados a diferentes disciplinas científicas (biología, geología, matemáticas, etc.).

1. Panorámica tecnologías de IA

Plataformas y herramientas

Hugging Face

Hugging Face ofrece el **Model Hub**, repositorio público con millones de modelos preentrenados (textuales, vision, multimodales) y conjuntos de datos. También proporciona **Spaces**, un directorio de aplicaciones de IA interactivas compartidas. La biblioteca *Transformers* de HF permite cargar, ajustar o servir estos modelos bajo PyTorch o TensorFlow. Por ejemplo, hay APIs para clasificación de texto, traducción, generación de imágenes, etc. HF dispone de planes gratuitos (por ejemplo: almacenar modelos públicos gratis) y servicios de inferencia pagos (~\$0.033/hora por endpoint dedicado), lo cual se detalla en su página de precios. Esta plataforma es muy usada en investigación y educación ya que facilita acceder a modelos sin grandes recursos propios.

LangChain

LangChain es un framework de código abierto (Python/JavaScript) diseñado para crear aplicaciones que combinan LLMs con fuentes de datos externas. Permite “encadenar” pasos: por ejemplo, recibe la consulta del usuario, la procesa o traduce si es necesario, luego realiza una búsqueda o recuperación de información relevante en bases de datos o documentos, y finalmente alimenta al LLM con ese contexto para generar una respuesta informada. Así, LangChain facilita construir agentes de IA o chatbots especializados (ej. un chatbot médico que primero busca en un repositorio de investigaciones antes de responder). Es modular y extensible, y su propósito educativo es mostrar cómo integrar datos reales en aplicaciones basadas en LLMs.

Google Colab

Google Colab es un servicio de **notebooks Jupyter en la nube** (no hay que instalar nada local) que combina código ejecutable, texto enriquecido e imágenes en un mismo documento. Proporciona un entorno “listo para usar” con Python y bibliotecas de ML preinstaladas, facilitando la colaboración entre usuarios (compartición de notebooks vía Google Drive). Una ventaja clave es que Colab regala cómputo acelerado: acceso gratuito a GPUs (por ejemplo NVIDIA Tesla T4) y TPUs (Google’s Tensor Processing Units). Esto permite entrenar o inferir modelos sin invertir en hardware costoso. Colab tiene límites de uso (tiempo de sesión, RAM) en el plan gratuito, y ofrece versiones *Pro* de pago para mayor capacidad. Por su sencillez, es muy popular en educación científica, permitiendo a profesores y alumnos ejecutar experimentos de IA sin infraestructura

local.

Figura: Interfaz conceptual de Google Colab con asistencia de IA. Colab combina código ejecutable y texto en notebooks colaborativos accesibles desde el navegador. Permite compartir trabajo con estudiantes y tiene asistentes de IA para autocompletar código.

Kaggle Notebooks

Kaggle (plataforma de ciencia de datos de Google) también ofrece notebooks colaborativos en la nube. Al igual que Colab, proveen GPUs gratuitas (ej. NVIDIA Tesla P100) y un entorno de Python listo, pero están más orientados a análisis de datos (con acceso directo a datasets de la comunidad) y a la competencia en proyectos de ciencia de datos. Son gratuitos y fáciles de usar para docentes que guían prácticas de análisis de datos con IA.

AWS SageMaker y Azure ML

Son **servicios en la nube** de AWS (Amazon) y Microsoft respectivamente. Permiten el entrenamiento, ajuste, despliegue y monitorización de modelos de ML a escala industrial. Por ejemplo, Azure Machine Learning es “un servicio en la nube que acelera y administra el ciclo de vida del aprendizaje automático”. Ofrece notebooks integrados, interfaces gráficas (Designer) y soporte para frameworks abiertos (PyTorch, TensorFlow, Scikit-learn). SageMaker similarmente. Ambos requieren pago según recursos usados (CPUs/GPUs en la nube), por lo que su uso educativo suele limitarse a instituciones con presupuesto o al plan gratuito inicial. Su ventaja es la escalabilidad y la integración con servicios empresariales (bases de datos, web services), aunque a costa de mayor complejidad y costo.

Ollama y LM Studio

Estas son **herramientas locales** para correr modelos de lenguaje en PCs sin depender de nube. *Ollama* (CLI/API) e *LM Studio* (aplicación de escritorio) permiten descargar y ejecutar LLMs open-source en Windows, Mac o Linux. Son gratuitas para uso personal. Por ejemplo, LM Studio anuncia: “Run AI models, locally and privately”, soportando modelos como gpt-oss, Qwen3 o Gemma3 en tu propio hardware. La figura ilustra su interfaz tipo IDE para inferir con un modelo en local. Al usarlos, los datos permanecen en tu dispositivo, lo que protege la privacidad. Estas plataformas son útiles en entornos educativos donde no se desea o no se puede usar la nube.

Figura: Captura esquemática de LM Studio ejecutando un modelo de IA localmente. LM Studio es una app gratuita que permite correr LLMs en el ordenador, incluyendo asistentes de autocompletado. Al usarse localmente, mantiene los datos del usuario fuera de la nube.

vLLM

vLLM es una biblioteca de inferencia desarrollada inicialmente en la Universidad de Berkeley (ahora con soporte de NVIDIA) diseñada para servir LLMs con **gran rendimiento y eficiencia de memoria**. Permite procesar muchas solicitudes en paralelo, optimiza la gestión de memoria de las llaves-valor de atención, y utiliza técnicas de cuantización (INT4/INT8, FP8). Se integra con modelos de Hugging Face y soporta GPUs (NVIDIA, AMD, Intel) e incluso CPUs, y hasta TPUs mediante plugins. vLLM ofrece un servidor API compatible con OpenAI para inferencia, ideal para montar demos educativas de IA de forma eficiente.

Bibliotecas principales

- **PyTorch:** Es una *biblioteca de tensores* optimizada para aprendizaje profundo sobre GPUs/CPUs. Ofrece un modo de ejecución ansioso (fácil de usar en investigación) y herramientas para producción (TorchScript, TorchServe). PyTorch cuenta con un ecosistema rico (TorchVision, TorchAudio) para visión, audio y NLP, y está soportado por principales nubes (AWS, GCP, Azure).
- **TensorFlow:** Es una plataforma de Google de extremo a extremo para ML. La página oficial destaca que “TensorFlow facilita la creación de modelos de AA que pueden ejecutarse en cualquier entorno”. Posee API intuitivas (como Keras) y bibliotecas auxiliares (TFX, TensorFlow Lite para móviles, etc.). TensorFlow también permite exportar modelos a otros formatos (SavedModel, TFLite) para desplegarlos en servidores o dispositivos móviles.
- **Scikit-learn:** Biblioteca de Python de propósito general para aprendizaje automático clásico. Brinda implementaciones eficientes de algoritmos de regresión, clasificación, clustering y reducción de dimensionalidad. Como señala IBM, “*Scikit-learn es una de las bibliotecas de machine learning más utilizadas*” en Python. Aunque no está diseñada para redes neuronales profundas, es indispensable para tareas de ciencia de datos (preprocesamiento, validación cruzada, métricas) en educación.

1. Panorámica tecnologías de IA

Técnicas de IA avanzadas

- **RAG (Retrieval-Augmented Generation):** técnica donde un LLM consulta una base de conocimientos externa antes de generar su respuesta. El modelo inserta información recuperada en su prompt o contexto, lo que mejora la precisión y actualidad de la respuesta. AWS explica que RAG optimiza la salida de un LLM “*refiriéndose a una base de conocimientos autorizada fuera de sus datos de entrenamiento*”. Es útil para aplicaciones donde se requiere citar fuentes o manejar información dinámica (chatbots basados en documentos, FAQ inteligentes, etc.).
- **Fine-tuning:** reentrenar un modelo preentrenado con un nuevo conjunto de datos específico. Por ejemplo, ajustar un LLM general para que sea experto en biología entrenándolo adicionalmente sobre textos científicos. Esto mejora el desempeño en la tarea objetivo sin entrenar desde cero.
- **Prompt engineering:** consiste en diseñar con cuidado los prompts (indicaciones) al modelo para obtener las respuestas deseadas. Aunque aún falta estandarización, es esencial formular preguntas precisas, incluir ejemplos o estructuras específicas para guiar al LLM.
- **LoRA (Low-Rank Adaptation):** método de ajuste de modelos muy grande que añade *matrices de bajo rango* adicionales en las capas de atención del modelo. Solo estas matrices se entrenan, mientras los pesos originales permanecen fijos. Esto reduce enormemente la cantidad de parámetros a ajustar (buena eficiencia de memoria) y evita que el modelo *olvide* lo preentrenado. En práctica, permite fine-tuning de modelos gigantes incluso en GPUs modestas (por ejemplo Tesla T4 o RTX 3080).
- **PEFT (Parameter-Efficient Fine-Tuning):** término general para técnicas como LoRA, *prefix tuning*, *prompt tuning*, etc., que sólo ajustan unos pocos parámetros adicionales. Como señala Hugging Face, PEFT congela la mayoría de los parámetros del modelo base y entrena sólo un pequeño subgrupo, abaratando el cómputo y el almacenamiento. Los resultados suelen ser comparables a un fine-tuning completo, pero con checkpoints muy pequeños (unos pocos MB en lugar de decenas de GB).

1. Panorámica tecnologías de IA

Infraestructura de IA

El entrenamiento y la inferencia de modelos IA requieren hardware acelerado. A continuación se compara los principales:

Tipo de hardware	Características/uso	Rendimiento relativo	Costo/ejecución	Accesibilidad educativa
GPU (NVIDIA)	Procesador paralelo general. Optimizado para matrices (CUDA, Tensor Cores). Soporta PyTorch, TF.	H100/A100: ~1-3 PFLOPS (FP16) por unidad. Gran VRAM (80-141GB). Soporta batch grande y redes de atención extensas.	Alto: \$4-10/h (GPU en nube). Tarjetas PC ~\$800-\$3000 según modelo.	Muy accesibles: Colab/Kaggle ofrecen GPUs gratis; muchas universidades usan GPUs gaming.
TPU (Google Cloud)	ASIC tensor específico. Integración fuerte con TensorFlow/JAX. Diseñado para inferencia y entrenamiento de ML en la nube. No disponible fuera de Google Cloud.	v6e: ~2 PFLOPS FP16 por chip. Masivo paralelismo (bajo costo por token).	Pago por uso: ~\$2.70/h por TPU v6e (nube Google). No hay versión local; uso sólo en servicios Google (Cloud TPU o Colab TPU gratuita).	Limitado: Colab da pequeñas TPUs gratis; uso educativo real en nube (p.ej. Google Cloud for Education créditos).
NPU / Neural Engine	Unidades IA en chips de móviles/PCs (ex. Apple, Huawei). Muy eficientes energéticamente. Se usan en visión, NLP en dispositivo.	Ej.: Apple ANE v5 (A15): 15.8 TFLOPS (FP16). La primera ANE (A11) fue 0.6 TFLOPS; cada generación crece mucho.	Integrado en dispositivos (smartphone/tablet). No se compra separado. Costo = el dispositivo (iPhone/AirPods/Mac con M-series).	Alta: Los estudiantes llevan móviles con NPU. Google Coral (Edge TPU) ~\$75 es asequible para demos de edge.
FPGA	Hardware reconfigurable (p.ej. Xilinx). Puede diseñarse el circuito específico para IA.	Rendimiento moderado. Menos paralelo que GPU en FP, pero baja latencia.	Alto de entrada: tarjetas FPGA avanzadas ~miles USD.	Bajo: Difícil de programar (Verilog) en cursos básicos; se usa más en investigación/industria. Existen kits educativos (Digilent) pero limitados.

Tipo de hardware	Características/uso	Rendimiento relativo	Costo/ejecución	Accesibilidad educativa
ASIC (EdgeTPU)	Chips específicos para IA (ej. Google Edge TPU, USB accelerator). Ultraeficientes para inferencia puntual.	Edge TPU (Google): ~4 TOPS/W. Rendimiento limitado a modelos pequeños (p.ej. MobileNet, BERT pequeño).	Moderado: Edge TPU USB ~\$75. Otros ASIC (Graphcore IPU, Habana) solo en servidores costosos.	Bueno: Edge TPUs para IoT / educación (Raspberry Pi + Coral). TPU/ASIC empresariales no disponibles en escuela.
Neuromórficos	Chips de investigación (Intel Loihi, IBM TrueNorth). Imitan redes neuronales físicas spiking.	Aún experimentales. Muy bajo consumo (ej. mil millones de OPS por segundo gastando milivatios).	<i>Experimental.</i> No comercial generalizada.	Muy bajo: solo en laboratorios especializados.

En resumen: las **GPU** son el estándar ampliamente usado (fáciles de acceder en colabs, PCs propias o nubes académicas). Las **TPU** ofrecen mayor eficiencia por coste en cargas de inferencia, pero sólo están en Google Cloud (aunque Colab da acceso limitado). Los **NPUs** son útiles para IA en móviles y dispositivos embebidos, mejorando privacidad y energía. FPGAs y ASICs sirven para casos muy particulares, no tan comunes en entornos educativos. Los aceleradores neuromórficos son aún investigación.

Además, como muestra la comparación de [49], GPUs (p.ej. NVIDIA H100/H200) tienen más VRAM y mejor soporte software (CUDA/PyTorch), mientras que TPUs se especializan en cargas TensorFlow con alta eficiencia. Por ejemplo, la H100 entrega ~150 tokens/s para LLaMA-70B con vLLM en AWS (mayor throughput), mientras que un TPU v6e puede dar ~120 tokens/s con TensorFlow pero con sólo 32 GB de memoria, necesitando 8 chips para LLaMA-70B.

Dónde ejecutar modelos: nube vs local vs edge

- **Nube:** Plataformas como Colab, AWS, Azure facilitan la puesta en marcha sin instalar nada. Se escala según demanda pero requiere conexión y genera costos (por cómputo/tiempo). Útil para demos en clase o proyectos que necesitan GPUs fuertes puntualmente.
- **Local (on-premises):** Ejecutar modelos en PCs, laptops o servidores propios. Ventaja en control de datos (privacidad) y sin latencia de red. Limitación en recursos de hardware: típicamente sólo CPUs o GPU de escritorio (RTX/Pascal/Turing) y menor RAM que un servidor. A menudo viable para inferencia en modelos medianos o entrenamiento ligero.

- **Edge (dispositivos):** Modelos corriendo en smartphones, IoT o dispositivos embebidos. Ventajas: latencia ultrabaja, privacidad total (datos no salen del aparato), operación offline. Desventajas: recursos muy limitados (se usan NPUs con poca memoria). Como indica el caso de Multiverse, comprimir modelos para edge puede democratizar IA («Edge Computing: enable AI on resource-limited devices, reducing cloud reliance»). En educación, esto se ve en proyectos que usan móviles o Arduino/NPU (p.ej. reconocimiento de imágenes on-device).

1. Panorámica tecnologías de IA

Fuentes de datos

Los datos son la base de la IA. Es fundamental usar **datasets de calidad** y bien documentados. Fuentes típicas incluyen repositorios académicos (Imagenet, COCO, UCI ML), colecciones científicas (GenBank para biología, EarthExplorer para geología, etc.) y plataformas como Hugging Face Datasets o Kaggle Datasets, que agrupan datos populares de múltiples dominios. En educación de ciencias, es interesante explorar datasets abiertos de áreas específicas (genómica, geolocalización, fórmulas matemáticas, espectros de química, etc.).

Hay que considerar la **ética y gobernanza**: respetar licencias y privacidad de datos (p.ej. evitar datos personales sin consentimiento), balancear representatividad (reducir sesgos), y documentar las fuentes. Muchos proyectos de IA académicos ahora incluyen tarjetas de datos (“datasheets”) que explican qué contiene y qué sesgos potenciales hay. Asimismo, organismos como la UE promueven iniciativas para gobernar la IA responsablemente, algo a destacar en clases de ética de la tecnología.

1. Panorámica tecnologías de IA

Tablas comparativas

Tablas comparativas

Plataformas y herramientas (funcionalidades, costo aproximado, facilidad de uso):

Plataforma/Herramienta	Funcionalidades principales	Coste aproximado	Facilidad de uso (docentes)
Hugging Face	Repositorio de modelos preentrenados , demos (Spaces), APIs de inferencia. Pipelines para NLP, visión, etc. Integración con PyTorch/TF.	Gratis (repos públicos). Planes pagos para endpoints dedicados (~0.03 \$/h).	Alta: web intuitiva, ejemplos. Baja curva para descargar/inferir modelos. Se requiere algo de Python para entrenar/ajustar.
LangChain	Framework para encadenar LLM con datos externos. Construye flujos de QA, agents, retrieval.	100% open-source (gratis). Dependencia: usa LLM (API de pago externa si se desea).	Media: requiere conocimientos de programación Python. Útil para proyectos avanzados de IA.
Google Colab	Notebook Jupyter en nube. Integración Drive, colaboración en tiempo real. Entornos preconfigurados con IA (Python, TF, PyTorch). GPU/TPU gratuitas.	Gratis (nivel básico). Plan Colab Pro ~\$10/mes para más recursos.	Muy alta: basta cuenta Google. Interfaz intuitiva para estudiantes. No requiere instalación.
Kaggle Notebooks	Similar a Colab; notebooks en nube con dataset integrado (Kaggle Datasets). GPUs gratuitas (Tesla P100) durante tiempo limitado. Enfoque datos + ML.	Gratis (alto cómputo). Competencias y dataset store sin costo.	Alta: integrado en sitio Kaggle. Aprendizaje de ML con ejemplos reales.

Plataforma/Herramienta	Funcionalidades principales	Coste aproximado	Facilidad de uso (docentes)
AWS SageMaker	Plataforma ML profesional. Notebooks, entrenamiento distribuido, endpoints de inferencia escalables. Integración AWS (S3, Lambda, etc.).	Pago por uso: instancias GPU (p.ej. p3.2xlarge ~\$3/h), almacenamiento. Tarifa variable.	Baja: complejo, requiere gestión de AWS. Para cursos avanzados o instituciones con soporte cloud.
Azure ML Studio	Servicio ML en Azure. Studio web con designer, notebooks, pipelines MLOps. Modelos pre-entrenados de Azure disponibles.	Pago por uso similar a AWS. Créditos gratuitos limitados.	Media-baja: entorno GUI amigable, pero concepto MLOps avanzado. Adecuado para proyectos organizacionales.
Ollama	CLI/API para correr LLM locales. Soporta descarga automática de modelos open-source (gemma, qwen, etc.).	Gratis (local).	Alta: solo descarga e instalación. Interfaz por CLI simple. Ideal para demos offline.
LM Studio	App de escritorio (Windows/Mac/Linux). Interfaz tipo IDE para LLMs locales (gpt-oss, Qwen3, etc.). Compatible con notebooks.	Gratis (uso personal/educativo).	Muy alta: GUI intuitiva. Fácil de instalar. No requiere nube, por lo que es apropiado para aulas sin internet.
vLLM	Biblioteca para inferencia LLM optimizada. Se integra con Transformers, soporta batching, cuantización y GPUs/CPU's múltiples.	Gratis (open-source).	Media: orientada a desarrolladores. Útil para crear servidores de inferencia en clase de IA.

Hardware (tipo, rendimiento relativo, coste/ejecución, accesibilidad educativa):

Acelerador / Hardware	Descripción / Uso	Rendimiento relativo	Coste / Ejecución	Accesibilidad educativa
GPU - NVIDIA (p.ej. H100/A100)	Procesador paralelo general; esencial para DL. Diseño genérico (CPU+Tensor Cores).	Muy alto para IA: H100 ~3.5 PFLOPS FP16, A100 ~1.5 PFLOPS.	\$\$\$\$ (e.g. H100 ~\$4.5/h en nube). Tarjeta A100 ~\$100k, H100 ~\$30k.	Alto: GPUs de juego (RTX 30xx/40xx) ~mil€/unidad. Accesibles en PCs, laboratorios. Colab/Kaggle ofrecen GPUs gratis.

Acelerador / Hardware	Descripción / Uso	Rendimiento relativo	Coste / Ejecución	Accesibilidad educativa
TPU (Google Cloud)	ASIC dedicado a tensores. Excelentes para inferencia TensorFlow, disponible en nube (v5, v6). Integrado solo con GCP.	v6e ~2 PFLOPS (FP16) por chip.	\$\$ (Google Cloud: ~\$2.70/h por v6e).	Limitado: no instalable en local. Colab ofrece uso limitado gratis. Google Cloud ofrece becas académicas.
NPU (Neural Processing Unit)	Aceleradores IA en móviles/SoC (Apple M-series, Huawei Kirin, etc.). Muy eficientes en consumo.	Ejemplo: Apple ANE v5 (A15): 15.8 TFLOPS (FP16). Rendimiento multi-TOPS con muy bajo consumo.	NPU ya incluido en costo del dispositivo. P.ej. iPhone/iPad/Mac (~\$700+).	Muy alto: cualquier smartphone reciente tiene NPU (Apple/A14+, Android con NPUs). Kits como Google Coral (~\$75) permiten demos de edge AI.
FPGA	Chips reprogramables (Xilinx, Intel). Se puede diseñar hardware para IA, pero es complejo.	Depende del diseño. Normalmente menor paralelismo en FP que GPU, pero baja latencia.	Alto: Board FPGA industrial ~\$300-\$1000.	Bajo: Requiere hardware especial y conocimientos de HDL. Poco común en clases básicas de IA.
ASIC IA	Chips fijos diseñados para IA (ej. Edge TPU de Google, IPU de Graphcore). Son muy eficientes para modelos concretos.	Google Edge TPU USB ~4 TOPS. Otros ASIC (TPUv5, Habana) llegan a PFLOPS.	Moderado: Edge TPU dev ~\$75. ASIC de centro de datos inaccesibles (~decenas de miles USD).	Medio: Edge TPUs y kits (Raspberry+Coral) permiten prácticas IA local. ASIC de centro no.
Neuromórfico	Chip experimental emulando neuronas (Intel Loihi, IBM TrueNorth). Potente en eficiencia energética (OPS/W).	Muy bajo consumo (por ejemplo, Loihi 2: miles de millones de spike-s OPS con décimas de vatio).	Experimental, no comercial.	Prácticamente ninguno: solo laboratorios de investigación con acceso restringido.

2. Fundamentos de Machine Learning y Deep Learning

2. Fundamentos de Machine Learning y Deep Learning

Aprender para y sobre la IA

4. Prompting, RAG, Fine Tuning

Configuración y propuestas ya creadas en las plataformas

4. Prompting, RAG, Fine Tuning

Herramientas y plataformas

5. Uso de agentes y automatización

Actividades paso a paso

5. Uso de agentes y automatización

Actividad 1

6. Aplicaciones de IA en el ámbito científico

6. Aplicaciones de IA en el ámbito científico

Casos de uso Generales

6. Aplicaciones de IA en el ámbito científico

Casos de uso en Biología y Geología

6. Aplicaciones de IA en el ámbito científico

Casos de uso en Física y Química

6. Aplicaciones de IA en el ámbito científico

Casos de uso en Matemáticas