1.Panorámica tecnologías de IA

- 1.1 Introducción y resumen del curso
- 1.2 IA Generativa, Modelos de IA y Agentes
- 1.3 Plataformas y herramientas
- 1.5 Infraestructura de IA y ejecución de modelos
- 1.6 Fuentes de datos

Copyright 2026

1.1 Introducción y resumen del curso

Este curso pretende fundamentalmente dos cosas, por un lado **profundizar en los fundamentos y tecnologías asociadas a los modelos de lenguaje para poder usarlos y aplicarlos según nuestras necesidades y por otro ofrecer al docente ideas y situaciones de aprendizaje para aplicar la IA en el ámbito científico-tecnológico**

Veremos en que se basa la IA, los fundamentos del *machine learning* y el aprendizaje supervisado así como la base del *deep learning* y su arquitectura más importante llamada *transfomers* que han dado lugar a los llamados modelos de lenguaje o LLMs, su diversidad y su forma de uso, no solo como las IAs tradicionales accediendo a través de una web, sino **usándolos en nuestro propio equipo de manera local permitiendo la confidencialidad de los datos y la independencia de terceros**. También veremos como influye el *hardware* o los equipos que usamos y cuales son más adecuados según los objetivos y aplicaciones.

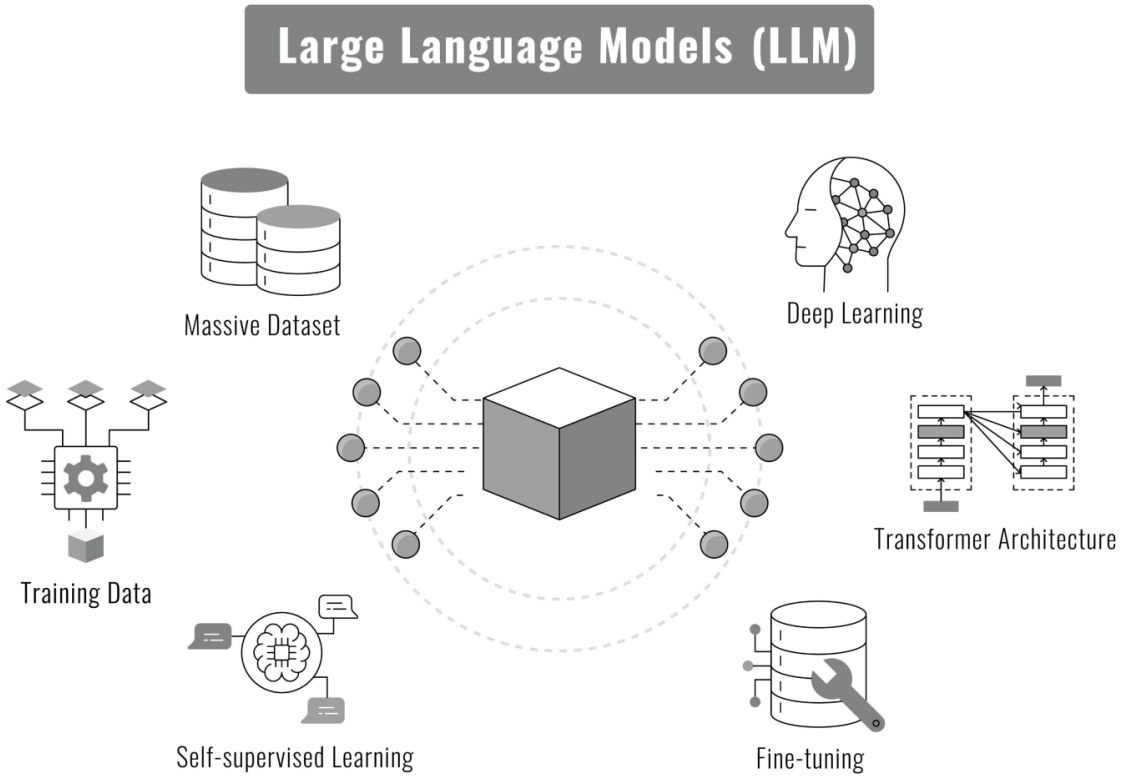
El enorme desarrollo y avance de las tecnologías de IA en los últimos meses está permitiendo que cada vez más gente ajena al mundo tecnológico tenga acceso a la posibilidad de realizar tareas que hace solamente dos o tres años requerían de una formación compleja y extensa en el tiempo. Esto hace que con pocos conocimientos ya seamos capaces de **desarrollar aplicaciones adaptadas a nuestro contexto académico** y sin uso de código, aunque también veremos como manejas nuestros modelos de IA usando lenguajes como *python* que hoy en día simplifican enormemente la programación haciéndola mucho más intuitiva y cercana al lenguaje humano.

También veremos tecnologías relacionadas y disruptivas que están surgiendo en los últimos meses, como son el **uso de agentes y automatización de procesos** mediante la combinación de modelos de lenguaje y diversas herramientas y aplicaciones como el correo electrónico o la búsqueda en *internet*.

Una vez consolidada esta **base técnica,** el curso se desplaza hacia la aplicación práctica en las asignaturas científico-técnicas. Aquí, la IA deja de ser el objeto de estudio para convertirse en una aliada estratégica en el laboratorio y el aula. Exploraremos cómo estas herramientas pueden modelar y **simular fenómenos físicos, asistir en la escritura de código complejo, analizar grandes volúmenes de datos experimentales o incluso generar simulaciones químicas y biológicas que antes requerían un software altamente especializado**. No buscamos ofrecerte un manual de instrucciones cerrado ni una receta única, sino proporcionarte una panorámica completa y versátil de las tecnologías disponibles. El objetivo final es que, al terminar, poseas tanto la **competencia técnica como el criterio pedagógico** para decidir exactamente

Copyright 2026

cómo integrar la IA en tu disciplina, ya sea para potenciar la capacidad de indagación de tus alumnos o para revolucionar la enseñanza de la ciencia y la tecnología desde una perspectiva moderna, práctica y rigurosa.



Copyright 2026

1.2 IA Generativa, Modelos de IA y Agentes

Los **LLMs (Large Language Models)** son modelos de lenguaje de gran tamaño (como GPT, BERT o LLaMA) entrenados con enormes corpus de texto para realizar tareas como generación de texto, traducción, clasificación o respuesta a preguntas. Plataformas como **Hugging Face** ofrecen bibliotecas como *Transformers*, que permiten gestionar y utilizar estos modelos en diferentes modalidades. Aunque originalmente estaban orientados al procesamiento de texto, hoy en día muchos de estos modelos pueden trabajar también con **imágenes, audio o vídeo**, dando lugar a sistemas **multimodales** capaces de combinar distintos tipos de información.

Además de los LLMs centrados en texto, existen **modelos de visión** basados en redes neuronales profundas, como las **CNN (Convolutional Neural Networks)** o los **Vision Transformers**, que se utilizan para tareas como clasificación de imágenes, detección de objetos o segmentación visual. También han aparecido modelos generativos como **Stable Diffusion**, que permiten crear imágenes a partir de descripciones textuales. En paralelo, se han desarrollado **modelos multimodales** (por ejemplo **CLIP** o **LLaVA**) capaces de relacionar texto e imagen, permitiendo tareas como describir una fotografía, responder preguntas sobre una imagen o generar contenido visual a partir de instrucciones en lenguaje natural.

Muchos de estos sistemas se construyen a partir de lo que se denomina un **modelo fundacional (foundation model)**. Se trata de grandes redes neuronales entrenadas previamente con enormes cantidades de datos generales —como los modelos GPT de OpenAI o BERT de Google— que aprenden patrones lingüísticos y conceptuales amplios. Estos modelos proporcionan una **capacidad general de comprensión y generación**, que posteriormente puede adaptarse a tareas específicas mediante técnicas como el **fine-tuning** o el ajuste mediante instrucciones. Por ejemplo, un modelo fundacional entrenado con grandes cantidades de texto puede posteriormente especializarse para responder preguntas médicas, analizar documentos legales o ayudar en tareas educativas.

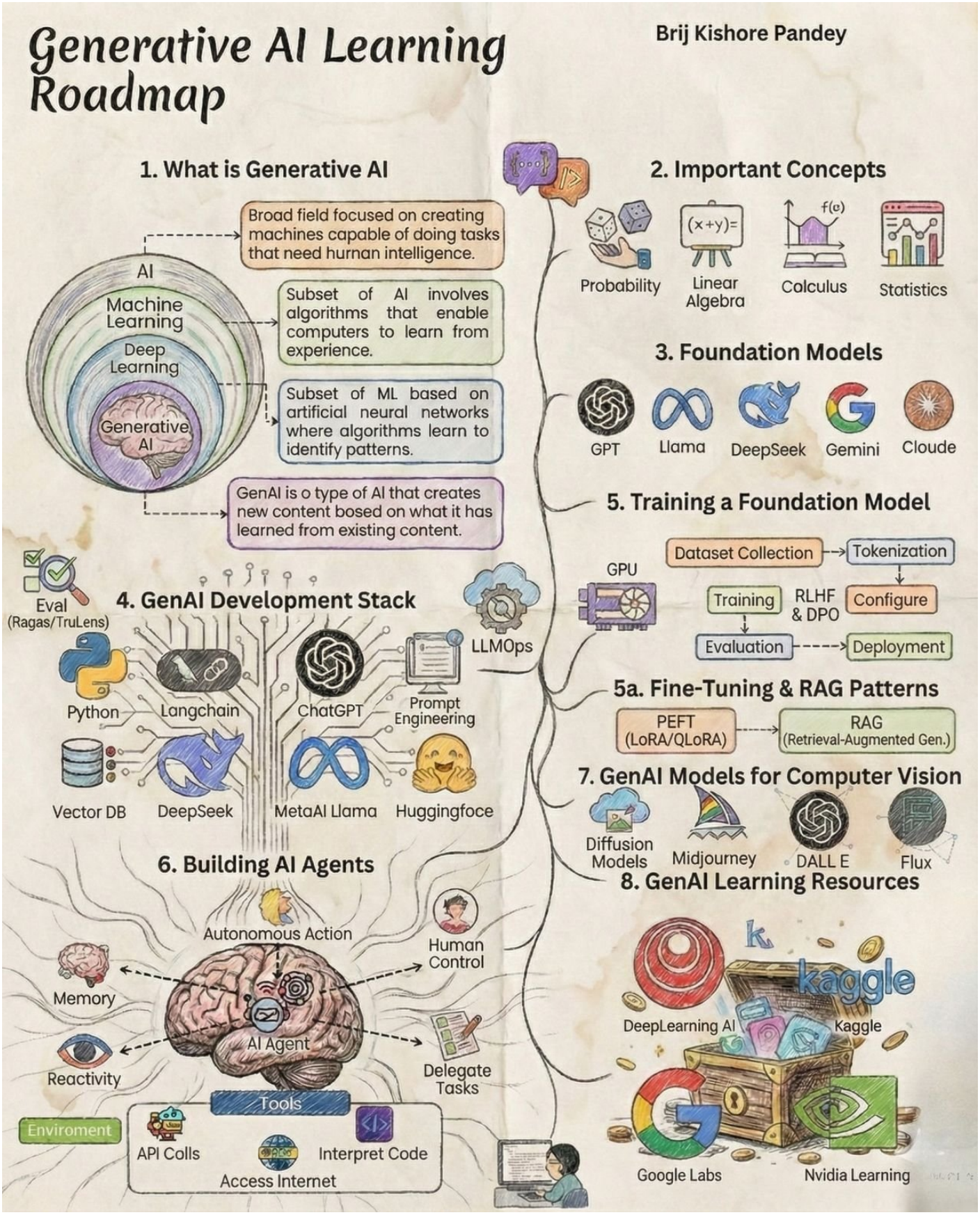
En los últimos años ha aparecido además un **nivel más avanzado de uso de los LLMs**: los **agentes de inteligencia artificial**. Un agente es un sistema que utiliza un modelo de lenguaje como núcleo de razonamiento, pero que además puede **tomar decisiones, planificar pasos y utilizar herramientas externas** (bases de datos, buscadores, programas o APIs) para resolver tareas complejas. En lugar de limitarse a generar una respuesta, el modelo puede descomponer un problema, ejecutar acciones y combinar distintos recursos para alcanzar un objetivo.

Copyright 2026

El desarrollo más reciente en este ámbito es la **orquestación de agentes**, donde varios agentes especializados colaboran entre sí dentro de un mismo sistema. En este enfoque, cada agente puede encargarse de una función concreta —por ejemplo, búsqueda de información, análisis de datos, generación de texto o verificación de resultados— y un sistema de coordinación organiza su interacción. Este paradigma está dando lugar a nuevas arquitecturas de software basadas en **equipos de agentes cooperativos**, capaces de automatizar procesos complejos y construir aplicaciones inteligentes más avanzadas.

En conjunto, la evolución de la inteligencia artificial ha pasado desde modelos especializados en tareas concretas hacia **modelos fundacionales generales**, después hacia **LLMs capaces de interactuar con múltiples modalidades de datos**, y finalmente hacia **sistemas de agentes y orquestación de agentes**, que representan actualmente una de las fronteras más activas de investigación y desarrollo en IA. Estos enfoques permiten que los modelos no solo generen contenido, sino que también **razonen, planifiquen y colaboren para resolver problemas reales en distintos ámbitos científicos, educativos y profesionales**.

Copyright 2026


Generative AI Learning Roadmap
Brij Kishore Pandey
1. What is Generative AI
Broad field focused on creating machines capable of doing tasks that need hurnan intelligence.
AI
Machine Learning
Deep Learning
Generative AI
Subset of AI involves algorithms that enable computers to learn from experience.
Subset of ML based on artificial neural networks where algorithms learn to identify patterns.
GenAI is o type of AI that creates new content bosed on what it has learned from existing content.
2. Important Concepts
Probability
Linear Algebra
Calculus
Statistics
3. Foundation Models
GPT
Llama
DeepSeek
Gemini
Cloude
4. GenAI Development Stack
Eval (Ragas/TruLens)
Python
Langchain
ChatGPT
Prompt Engineering
LLMOps
Vector DB
DeepSeek
MetaAI Llama
Huggingfoce
5. Training a Foundation Model
GPU
Dataset Collection
Tokenization
Training
RLHF & DPO
Configure
Evaluation
Deployment
5a. Fine-Tuning & RAG Patterns
PEFT (LoRA/QLoRA)
RAG (Retrieval-Augmented Gen.)
6. Building AI Agents
Autonomous Action
Human Control
Memory
AI Agent
Reactivity
Delegate Tasks
Tools
Enviroment
API Colls
Interpret Code
Access Internet
7. GenAI Models for Computer Vision
Diffusion Models
Midjourney
DALL E
Flux
8. GenAI Learning Resources
DeepLearning AI
Kaggle
Google Labs
Nvidia Learning

1.3 Plataformas y herramientas

En los últimos años ha habido un desarrollo exponencial de tecnologías vinculadas a la IA en todos los niveles y ámbitos profesionales

Este curso pretende dar una visión completa de las mismas e iniciar al alumno en un uso más profundo y flexible de la IA para adaptarla a sus necesidades y reducir la dependencia.

> **Aunque se habla de *python* y librerías asociadas no es en absoluto la idea del curso aprender a programar pero sí ofrecer dicha posibilidad a los alumnos que quieran introducirse en el tema.**

A continuación mostramos las principales y más populares plataformas y tecnologías. Todas ellas se usan en todo el mundo como referencia y base de pruebas y aprendizaje de IA con modelos y conjuntos de datos propios y ajenos

Modelos de IA

Entre los **modelos de IA más populares** se pueden distinguir varios grupos según el tipo de información que procesan y su aplicación principal.

Modelos de lenguaje (LLM)

Están diseñados para **comprender y generar texto**. Se utilizan en asistentes conversacionales, generación de contenido, programación asistida, traducción o análisis de documentos.
Ejemplos representativos:

- **GPT** (OpenAI): base de muchos sistemas conversacionales.
- **BERT** (Google): muy utilizado en comprensión del lenguaje y buscadores.
- **LLaMA** (Meta): modelo abierto que ha impulsado el ecosistema *open source*.
- **Mistral, Qwen o Phi**: modelos más ligeros que permiten ejecutar IA en local.

Modelos de visión por computador

Están diseñados para **analizar imágenes o vídeo**. Permiten tareas como reconocimiento de objetos, clasificación de imágenes o detección de patrones visuales.
Ejemplos:

Copyright 2026

- **CNN (Convolutional Neural Networks)**: arquitectura clásica para reconocimiento de imágenes.
- **Vision Transformers (ViT)**: aplican la arquitectura transformer al análisis visual.
- **Modelos de difusión** (como **Stable Diffusion**): generan imágenes a partir de texto.

Modelos de audio y voz

Procesan **sonido o lenguaje hablado**. Se utilizan para reconocimiento de voz, síntesis de voz o generación de música.
Ejemplos destacados:

- **Whisper** (OpenAI): transcripción automática de voz a texto.
- **WaveNet** (DeepMind): generación de voz natural.
- **MusicGen** o **AudioLM**: generación de música o audio mediante IA.

Modelos multimodales

Estos modelos pueden **combinar varios tipos de datos** (texto, imagen, audio, etc.). Permiten tareas como describir imágenes, responder preguntas sobre vídeos o interactuar con distintos tipos de contenido al mismo tiempo.
Ejemplos:

- **CLIP** (OpenAI): relaciona texto e imágenes.
- **LLaVA**: integra modelos de lenguaje y visión.
- **Gemini** o **GPT multimodal**: capaces de trabajar con múltiples modalidades de información.

Sistemas avanzados basados en LLM: agentes de IA

Una evolución reciente son los **agentes de inteligencia artificial**, que utilizan modelos de lenguaje como núcleo de razonamiento pero además pueden **planificar tareas, usar herramientas externas y ejecutar acciones**.

Cuando varios agentes especializados colaboran dentro de un mismo sistema coordinado se habla de **orquestación de agentes**, una de las tendencias más recientes en el desarrollo de aplicaciones avanzadas de inteligencia artificial.

Plataformas de desarrollo y pruebas de modelos

Existen diferentes tipos de plataformas para trabajar con inteligencia artificial, que van desde **entornos colaborativos en la nube** hasta **herramientas que permiten ejecutar modelos de forma local**.

Plataformas de modelos y repositorios de IA

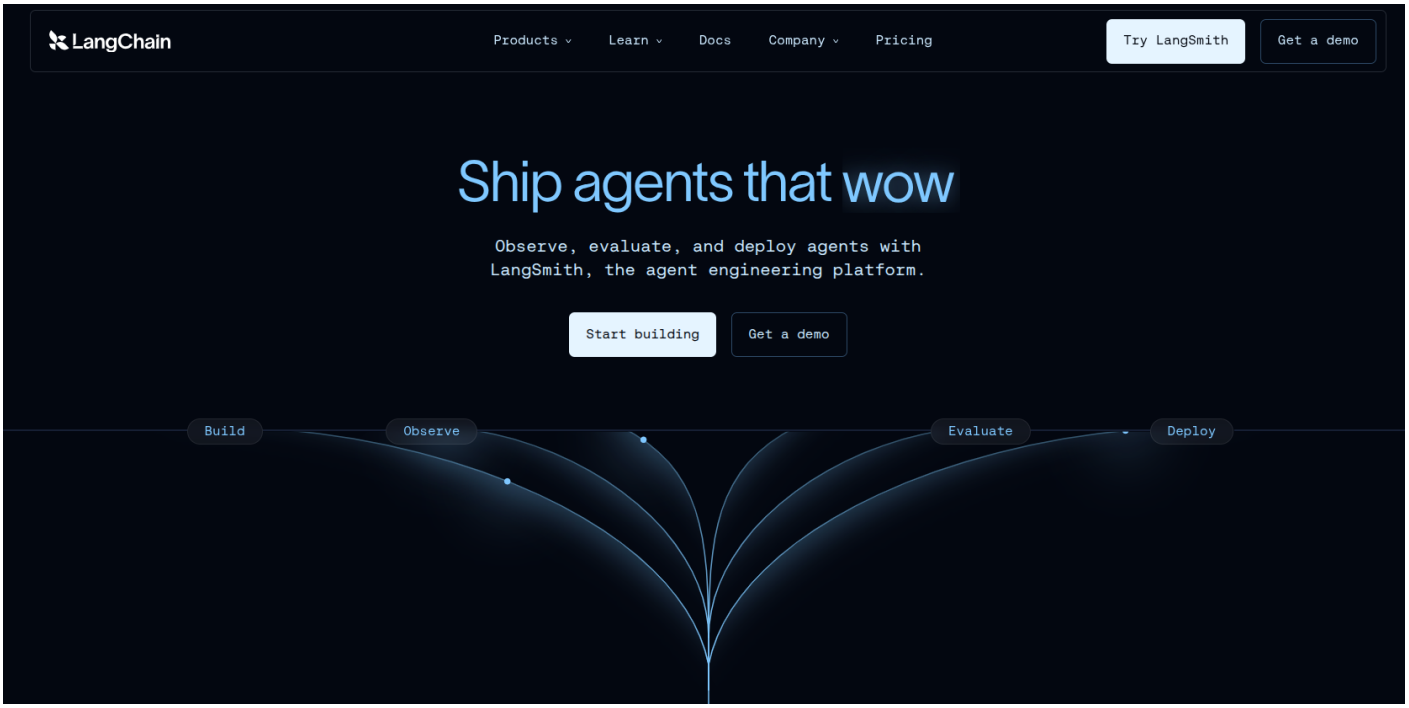
Son espacios donde investigadores y desarrolladores publican modelos, datasets y demos listas para usar.
Ejemplos:

- **Hugging Face**: uno de los mayores repositorios de IA del mundo, con millones de modelos, datasets y demos interactivos (*Spaces*).
- **ModelScope** o repositorios similares: plataformas que permiten descargar y probar modelos abiertos.

Frameworks para construir aplicaciones con LLMs

Facilitan la creación de aplicaciones complejas combinando modelos, bases de datos y herramientas externas.
Ejemplos:

- **LlamaIndex***: especializado en integrar datos propios con modelos de lenguaje.*
- **LangChain**: framework para conectar LLMs con fuentes de datos, APIs o bases de conocimiento.



Vista de la web de lancghain para programar, explorar y probar modelos

Entornos de desarrollo en la nube

Permiten experimentar con IA sin instalar nada en el ordenador, usando notebooks con acceso a GPU o TPU.
Ejemplos:

- ***Google Colab**: notebooks de Python en la nube con acceso sencillo a GPUs.*
- ***Kaggle Notebooks**: entorno similar, muy utilizado en ciencia de datos y competiciones de IA.*

Plataformas empresariales de Machine Learning

Ofrecen infraestructuras completas para entrenar, desplegar y gestionar modelos de IA en producción.
Ejemplos:

- ***AWS SageMaker**: entorno de desarrollo y despliegue de modelos en la nube de Amazon.*
- ***Azure Machine Learning**: plataforma de Microsoft para crear y operar sistemas de IA.*

Plataformas para ejecutar IA en local

Permiten trabajar con modelos abiertos sin depender de servicios en la nube, lo que mejora la privacidad y el control de datos.
Ejemplos:

- ***Ollama**: herramienta sencilla para ejecutar modelos open source en el ordenador.*
- ***LM Studio**: interfaz gráfica para descargar y usar LLMs locales.*

Motores de inferencia optimizados

Son herramientas especializadas para ejecutar modelos grandes de forma eficiente.
Ejemplos:

- ***vLLM**: motor optimizado para servir LLMs con alto rendimiento y bajo consumo de memoria, muy utilizado en servidores de IA.*

En conjunto, estas plataformas permiten **experimentar, desarrollar y desplegar aplicaciones de inteligencia artificial**, tanto en entornos educativos y de investigación como en sistemas profesionales a gran escala.

Técnicas clave

El uso práctico de los modelos de inteligencia artificial no depende solo del modelo en sí, sino también de una serie de técnicas que permiten adaptarlos, controlarlos o integrarlos con datos propios.

Prompting (ingeniería de instrucciones)

Consiste en formular correctamente las instrucciones que se dan al modelo para obtener mejores resultados. Incluye técnicas como zero-shot prompting (pedir una tarea directamente sin

ejemplos), few-shot prompting (mostrar algunos ejemplos para guiar la respuesta) y chain-of-thought prompting, donde se pide al modelo que razone paso a paso para mejorar la calidad de la respuesta.

RAG (Retrieval-Augmented Generation)

Es una técnica que permite a un modelo consultar información externa antes de generar la respuesta. En lugar de depender solo de lo aprendido durante el entrenamiento, el modelo puede buscar información en bases de datos, documentos o repositorios y utilizar esos datos como contexto. Se utiliza con frecuencia en sistemas de consulta sobre documentos, asistentes empresariales o aplicaciones de “hablar con tus datos”.

Fine-tuning (ajuste fino)

Consiste en volver a entrenar un modelo fundacional utilizando datos específicos para especializarlo en una tarea concreta. Por ejemplo, un modelo general puede ajustarse con documentos médicos, legales o educativos para mejorar su precisión en ese ámbito.

Embeddings y búsqueda semántica

Los modelos pueden transformar textos en representaciones numéricas llamadas embeddings que capturan el significado del contenido. Esto permite realizar búsquedas semánticas, encontrar documentos relacionados o agrupar información similar aunque no contengan exactamente las mismas palabras.

Uso de herramientas externas

Los modelos pueden conectarse con APIs, bases de datos, buscadores o programas externos para ampliar sus capacidades. De esta forma pueden consultar información actualizada, realizar cálculos o ejecutar acciones fuera del propio modelo.

Agentes de inteligencia artificial y orquestación de agentes

Una evolución reciente consiste en utilizar los modelos de lenguaje como agentes capaces de planificar tareas, tomar decisiones y utilizar herramientas de forma autónoma. En sistemas más avanzados, varios agentes especializados pueden colaborar entre sí bajo un sistema de coordinación, lo que se conoce como orquestación de agentes. Este enfoque permite construir aplicaciones de IA más complejas capaces de resolver tareas largas o procesos completos de trabajo.

Infraestructura y hardware

El desarrollo y uso de sistemas de inteligencia artificial requiere una infraestructura informática capaz de procesar grandes volúmenes de datos y ejecutar modelos complejos. Dependiendo del

Copyright 2026

tipo de aplicación, esta infraestructura puede ir desde ordenadores personales hasta centros de datos especializados.

Procesadores especializados

Los modelos de IA suelen ejecutarse en hardware optimizado para cálculos paralelos. Las unidades de procesamiento gráfico (GPU) se han convertido en el estándar para entrenar y ejecutar redes neuronales, ya que pueden realizar miles de operaciones simultáneamente. También existen otros aceleradores como las TPU (Tensor Processing Units) desarrolladas por Google o los chips especializados para IA integrados en dispositivos móviles.

Servidores y centros de datos

Las empresas que desarrollan modelos fundacionales suelen utilizar grandes centros de datos con miles de GPUs conectadas entre sí. Estas infraestructuras permiten entrenar modelos con billones de parámetros utilizando enormes cantidades de datos. Los centros de datos modernos también incluyen sistemas de almacenamiento de alto rendimiento y redes de alta velocidad para mover grandes volúmenes de información entre máquinas.

Computación en la nube

Muchos proyectos de inteligencia artificial utilizan servicios de computación en la nube que permiten acceder a hardware potente sin necesidad de comprarlo. Plataformas como AWS, Google Cloud o Microsoft Azure ofrecen instancias con GPU o TPU que se pueden alquilar por horas para entrenar o ejecutar modelos de IA.

Infraestructura local

Además de la nube, cada vez es más común ejecutar modelos de inteligencia artificial en equipos locales. Ordenadores personales con GPUs modernas pueden ejecutar modelos abiertos de tamaño medio, especialmente si están optimizados para uso local. Esto permite mayor control sobre los datos y reduce la dependencia de servicios externos.

Optimización y eficiencia

Debido al gran consumo de recursos de los modelos actuales, han surgido técnicas para mejorar su eficiencia, como la cuantización de modelos, la reducción de precisión o el uso de motores de inferencia optimizados. Estas técnicas permiten ejecutar modelos grandes en hardware más limitado, facilitando su uso en entornos educativos, dispositivos personales o aplicaciones empresariales.

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

Copyright 2026

Tipo de hardware	Características/uso	Rendimiento relativo	Costo/ejecución	Accesibilidad educativa
GP **U (NVIDIA)**	Procesador paralelo general. Optimizado para matrices (CUDAM, 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 gen 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.
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).	*Experimental.* 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

Copyright 2026

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).

Bibliotecas python

Para los que se animen a programar estas son las librerías más populares de amplio uso en el mundo del desarrollo de la IA

PyTorch

Es una biblioteca orientada al aprendizaje profundo que permite trabajar con tensores y entrenar redes neuronales utilizando CPU o GPU. Se caracteriza por su modo de ejecución flexible, muy utilizado en investigación, y por su ecosistema de herramientas especializadas como TorchVision para visión por computador o TorchAudio para procesamiento de audio. También ofrece utilidades para producción como TorchScript o TorchServe.

TensorFlow

Es una plataforma desarrollada por Google para crear y desplegar modelos de aprendizaje automático de forma completa. Incluye APIs sencillas como Keras para diseñar redes neuronales y herramientas adicionales para distintos entornos, como TensorFlow Lite para dispositivos móviles o TFX para sistemas de producción. Los modelos pueden exportarse en diferentes formatos para ejecutarse en servidores o dispositivos.

Scikit-learn

Es una de las bibliotecas más utilizadas en Python para aprendizaje automático clásico. Proporciona implementaciones de algoritmos de regresión, clasificación, clustering y reducción de dimensionalidad. Aunque no está diseñada para redes neuronales profundas, es muy utilizada en ciencia de datos para tareas de preprocesamiento, evaluación de modelos y experimentación educativa.

Copyright 2026

1.5 Infraestructura de IA y ejecución de modelos

Cuando trabajamos con inteligencia artificial —especialmente con modelos grandes como redes neuronales o modelos de lenguaje— el hardware se convierte en un elemento fundamental. Los cálculos necesarios para entrenar o ejecutar estos modelos son enormes y requieren procesadores capaces de realizar **millones o incluso billones de operaciones matemáticas por segundo**.

Por esta razón, en los últimos años han aparecido distintos tipos de procesadores especializados para IA. Mientras que los ordenadores tradicionales utilizan principalmente **CPUs**, el desarrollo de la inteligencia artificial ha impulsado el uso de **GPUs, TPUs y NPUs**, cada uno optimizado para distintos tipos de tareas.

De forma sencilla, podemos pensar en ellos como diferentes “motores” de cálculo diseñados para distintos contextos: centros de datos, investigación, ordenadores personales o dispositivos móviles.

CPU (procesador tradicional)
Las CPUs son los procesadores generales de los ordenadores. Son muy versátiles y pueden ejecutar todo tipo de programas, pero no están optimizadas para los cálculos masivos que requieren los modelos de inteligencia artificial.

Una CPU moderna puede realizar solo unas pocas operaciones simultáneas, mientras que los modelos de IA requieren realizar miles o millones de operaciones matemáticas en paralelo.

Por eso, aunque una CPU puede ejecutar modelos pequeños, no suele ser suficiente para entrenar redes neuronales grandes.

GPU (Graphics Processing Unit)
Las **GPUs** fueron diseñadas originalmente para gráficos y videojuegos, pero resultaron ser muy eficaces para inteligencia artificial.

Su gran ventaja es el **procesamiento paralelo**: una GPU puede tener miles de núcleos trabajando al mismo tiempo, lo que permite ejecutar muchas operaciones matemáticas simultáneamente.

Esto las convierte en la herramienta principal para entrenar modelos de deep learning. Hoy en día, la mayoría de los modelos de IA se entrenan utilizando GPUs en centros de datos o clusters de computación.

Copyright 2026

Ventajas principales de las GPUs

- Gran capacidad de cálculo paralelo
- Compatibilidad con muchos frameworks de IA (PyTorch, TensorFlow, JAX)
- Ecosistema de software muy desarrollado
- Flexibilidad para distintos tipos de tareas

Por esta razón, empresas como NVIDIA dominan gran parte del hardware utilizado para entrenar modelos de IA.

TPU (Tensor Processing Unit)
Las **TPUs** son procesadores diseñados específicamente para inteligencia artificial. Fueron desarrolladas por Google para acelerar operaciones matemáticas típicas de redes neuronales, especialmente multiplicaciones de matrices.

A diferencia de las GPUs, las TPUs utilizan una arquitectura especializada llamada **systolic array**, que permite realizar cálculos de forma extremadamente eficiente en tareas de aprendizaje profundo.

En algunos casos, las TPUs pueden ofrecer una eficiencia energética mucho mayor que CPUs y GPUs para tareas de inferencia y entrenamiento.

Ventajas principales de las TPUs

- Muy eficientes en operaciones de redes neuronales
- Alto rendimiento por consumo energético
- Integración directa con plataformas como Google Cloud

Sin embargo, suelen ser menos flexibles que las GPUs y están más orientadas a ciertos frameworks.

NPU (Neural Processing Unit)
Las **NPUs** son procesadores diseñados específicamente para ejecutar modelos de IA en dispositivos pequeños como móviles, ordenadores portátiles o dispositivos IoT.

Se encuentran, por ejemplo, en chips de teléfonos o en procesadores modernos de laptops (como Apple Silicon o Intel AI Boost). Su objetivo principal es ejecutar modelos de IA de forma eficiente directamente en el dispositivo.

Una de sus principales ventajas es la **eficiencia energética**, ya que consumen mucha menos energía que GPUs o CPUs para tareas de inferencia.

Copyright 2026

Esto permite ejecutar funciones de inteligencia artificial en tiempo real, como reconocimiento de voz, visión artificial o asistentes inteligentes, sin depender de la nube.

Ventajas principales de las NPUs

- Muy bajo consumo energético
- Ideales para dispositivos móviles o edge computing
- Buen rendimiento para inferencia en tiempo real

Tabla comparativa de tecnologías de hardware para IA

Tecnología	Tipo de dispositivo	Uso principal	Ventajas	Ejemplos
CPU	Ordenadores generales	Tareas generales, control del sistema	Gran flexibilidad	Intel Xeon, AMD EPYC
GPU	Centros de datos, PCs	Entrenamiento de modelos de IA	Gran paralelismo y ecosistema software	NVIDIA H100, A100, AMD MI300
TPU	Infraestructura cloud	Entrenamiento e inferencia de modelos grandes	Muy eficiente para operaciones de tensor	Google TPU v5, v6, v7
NPU	Móviles, laptops, edge devices	Inferencia local de IA	Muy bajo consumo energético	Apple Neural Engine, Intel AI Boost, Qualcomm Hexagon

La infraestructura de hardware es uno de los pilares del desarrollo de la inteligencia artificial. Mientras que las CPUs siguen siendo esenciales para tareas generales, el crecimiento de la IA ha impulsado el desarrollo de aceleradores especializados como GPUs, TPUs y NPUs.

Las GPUs dominan el entrenamiento de modelos grandes, las TPUs ofrecen una gran eficiencia en centros de datos y las NPUs permiten llevar la inteligencia artificial directamente a dispositivos personales.

En la práctica, los sistemas modernos suelen combinar varios de estos componentes, creando arquitecturas heterogéneas capaces de aprovechar lo mejor de cada tipo de procesador.

Dónde ejecutar modelos: nube, local y edge

Cuando trabajamos con modelos de inteligencia artificial, una de las decisiones importantes es **dónde se van a ejecutar**. En la práctica existen tres opciones principales: usar infraestructura en la nube, ejecutarlos en equipos propios o hacer que funcionen directamente en dispositivos. Cada opción tiene ventajas y limitaciones, y suele elegirse en función del tipo de proyecto, los recursos disponibles y el nivel de control que se necesita sobre los datos.

Copyright 2026

Computación en la nube
La nube es probablemente la forma más sencilla de empezar a trabajar con inteligencia artificial. Plataformas como Google Colab, AWS o Azure permiten ejecutar modelos sin instalar nada en el ordenador. El usuario simplemente abre un entorno en línea y puede utilizar recursos potentes, como GPUs o grandes cantidades de memoria.

La principal ventaja de la nube es que **permite escalar fácilmente el hardware según la necesidad**. Si un proyecto necesita mucha potencia de cálculo, se pueden utilizar servidores muy potentes durante unas horas o días. Esto es especialmente útil para entrenar modelos grandes o para proyectos que requieren GPUs avanzadas.

Otra ventaja es la facilidad de uso: muchas plataformas ya incluyen entornos de programación, bibliotecas y datasets preparados para trabajar.

Sin embargo, también tiene algunas limitaciones. La nube depende de una conexión a internet y normalmente implica costes asociados al tiempo de cálculo o al uso de recursos. Además, enviar datos a servidores externos puede generar preocupaciones relacionadas con la privacidad.

En educación, la nube es muy útil para **demos en clase, experimentos o proyectos puntuales que necesitan hardware potente**.

Infraestructura local (on-premises)
Otra posibilidad es ejecutar los modelos directamente en ordenadores propios, como PCs, portátiles o servidores del centro educativo o de la empresa. A esto se le llama infraestructura **local o on-premises**.

La principal ventaja de este enfoque es el **control total sobre los datos**. Como la información no sale del equipo o del servidor local, es más fácil mantener la privacidad y cumplir políticas de seguridad.

También se evita la latencia de red, es decir, el tiempo que tarda la información en viajar a servidores remotos.

El inconveniente principal es que los recursos de hardware suelen ser más limitados. Un ordenador personal normalmente dispone de menos memoria y menos potencia de cálculo que un servidor en la nube. Por eso, este enfoque suele utilizarse para **inferencias de modelos medianos o experimentos de entrenamiento ligero**.

En muchos entornos educativos o de investigación se utilizan PCs con GPUs de escritorio (por ejemplo tarjetas RTX) para ejecutar modelos open-source o experimentar con proyectos de IA.

Computación en el edge (dispositivos)
La tercera opción es ejecutar los modelos directamente en dispositivos como móviles, sensores,

Copyright 2026

cámaras inteligentes o microcontroladores. Este enfoque se conoce como **edge computing**.

La idea es que el procesamiento se realice **cerca del lugar donde se generan los datos**, en lugar de enviarlos a un servidor remoto. Esto tiene varias ventajas importantes.

La primera es la **latencia extremadamente baja**. Al procesar la información localmente, las respuestas pueden generarse casi en tiempo real, algo fundamental para aplicaciones que requieren decisiones rápidas.

Otra ventaja es la **privacidad**, ya que los datos sensibles pueden procesarse directamente en el dispositivo sin enviarse a la nube.

Además, los sistemas edge pueden funcionar incluso sin conexión a internet, lo que permite operar en entornos remotos o con conectividad limitada.

Sin embargo, los dispositivos edge tienen recursos limitados. Suelen disponer de menos memoria, menos almacenamiento y menor capacidad de cálculo que los servidores cloud.

Por esta razón, los modelos que se ejecutan en estos dispositivos suelen estar **comprimidos u optimizados** para funcionar con menos recursos.

En educación, este enfoque aparece en proyectos que utilizan teléfonos móviles, cámaras inteligentes o microcontroladores (como Arduino o Raspberry Pi) para ejecutar modelos de reconocimiento de imágenes o sonido directamente en el dispositivo.

Tabla comparativa: nube vs local vs edge

Entorno	Dónde se ejecuta	Ventajas	Limitaciones	Ejemplos de uso
Nube	Centros de datos remotos	Gran potencia de cálculo, escalabilidad, fácil acceso	Costes, dependencia de internet, privacidad	entrenamiento de modelos grandes, demos con GPUs
Local (on-premises)	PCs o servidores propios	Control de datos, sin latencia de red	hardware limitado	investigación, ejecución de modelos open-source
Edge	dispositivos y sensores	latencia muy baja, privacidad, funciona offline	recursos muy limitados	móviles, IoT, reconocimiento de imágenes en dispositivos

No existe una única solución válida para todos los casos. La nube ofrece potencia y escalabilidad, los sistemas locales ofrecen control y privacidad, y el edge permite ejecutar inteligencia artificial directamente en dispositivos con respuestas inmediatas.

Copyright 2026

Por eso, muchos sistemas actuales utilizan **arquitecturas híbridas**, donde el entrenamiento se realiza en la nube, mientras que la inferencia o el uso final del modelo se ejecuta en equipos locales o dispositivos edge.

Este enfoque combinado permite aprovechar lo mejor de cada entorno y es una de las tendencias más importantes en la infraestructura moderna de inteligencia artificial.

Copyright 2026

1.6 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.

Conjuntos de datos

Los modelos de inteligencia artificial se entrenan a partir de grandes colecciones de datos que contienen ejemplos del tipo de información que el sistema debe aprender a procesar. Estos **datasets** constituyen uno de los elementos fundamentales del desarrollo de la IA, ya que la calidad, diversidad y tamaño de los datos influyen directamente en el rendimiento del modelo.

Datos de texto

En el caso de los modelos de lenguaje, los conjuntos de datos suelen estar formados por **grandes colecciones de textos** procedentes de libros, artículos científicos, páginas web, código fuente o conversaciones. Estos datos permiten a los modelos aprender patrones del lenguaje, estructuras gramaticales y relaciones entre conceptos.

Datos de imagen

Los modelos de visión por computador se entrenan con **bases de datos de imágenes etiquetadas** que indican qué aparece en cada fotografía. Estos datasets permiten que los modelos aprendan a reconocer objetos, personas o escenas. Algunos conjuntos de datos muy conocidos incluyen millones de imágenes clasificadas en distintas categorías.

Datos de audio y voz

Los sistemas de reconocimiento o generación de voz utilizan **grabaciones de audio acompañadas de transcripciones**. Con estos datos los modelos aprenden a relacionar sonidos

Copyright 2026

con palabras o a generar voz sintética con características naturales.

Datos multimodales

En los últimos años se han desarrollado datasets que **combinan diferentes tipos de información**, como imágenes con descripciones en texto, vídeos con subtítulos o audio con anotaciones. Estos conjuntos de datos permiten entrenar modelos capaces de comprender varias modalidades de información al mismo tiempo.

Datos especializados o de dominio

Además de los grandes datasets generales, también existen **conjuntos de datos específicos para un ámbito concreto**, como medicina, derecho, finanzas o educación. Estos datasets se utilizan para especializar modelos mediante técnicas como el fine-tuning o para construir sistemas de consulta basados en RAG.

Calidad y preparación de los datos

Antes de utilizar un dataset en un modelo de IA, los datos suelen pasar por procesos de **limpieza, filtrado y anotación** para eliminar errores, duplicados o información irrelevante. También es importante considerar aspectos como el sesgo de los datos, la privacidad y los derechos de uso, ya que estos factores pueden influir en el comportamiento y la fiabilidad de los sistemas de inteligencia artificial.

Fuentes de datasets

Existen diversas plataformas donde investigadores y desarrolladores pueden encontrar conjuntos de datos para proyectos de inteligencia artificial. Algunas de las más utilizadas son **Kaggle**, que ofrece miles de datasets y competiciones de ciencia de datos; **Google Dataset Search**, un buscador especializado en localizar conjuntos de datos públicos en internet; y repositorios como **Hugging Face**, que además de modelos incluye colecciones de datasets preparados para entrenamiento y experimentación.

Copyright 2026