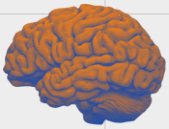
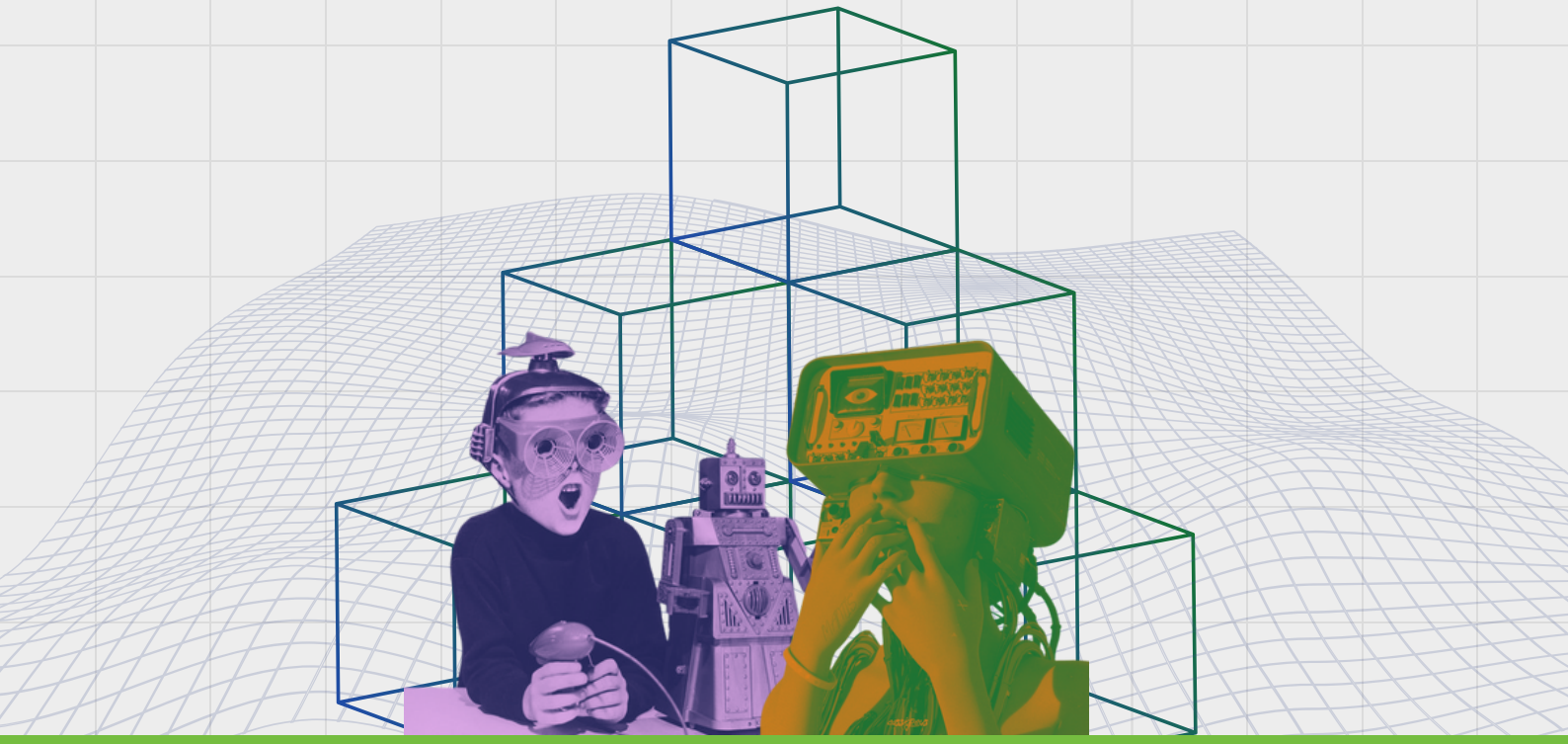
PROGRAMA DE INTELIGENCIA ARTIFICIAL



UNA INTRODUCCIÓN A LA

# INTELIGENCIA ARTIFICIAL

## DE LAS PRIMERAS NEURONAS ARTIFICIALES A LOS MODELOS DE LENGUAJE: EL RECORRIDO DE LA IA



UNIVERSIDAD NACIONAL DE HURLINGHAM

SECRETARÍA ACADÉMICA
INSTITUTO DE TECNOLOGÍA E INGENIERÍA



Universidad Nacional de Hurlingham -
Una introducción a la Inteligencia Artificial: De las primeras neuronas artificiales a los modelos de lenguaje: el recorrido de la IA - Hurlingham, 2025
Cuadernillo digital, PDF.

Autora: Clara Campos / CONICET - UNAHUR

Revisión de estilo asistida con herramientas de inteligencia artificial (ChatGPT, OpenAI).

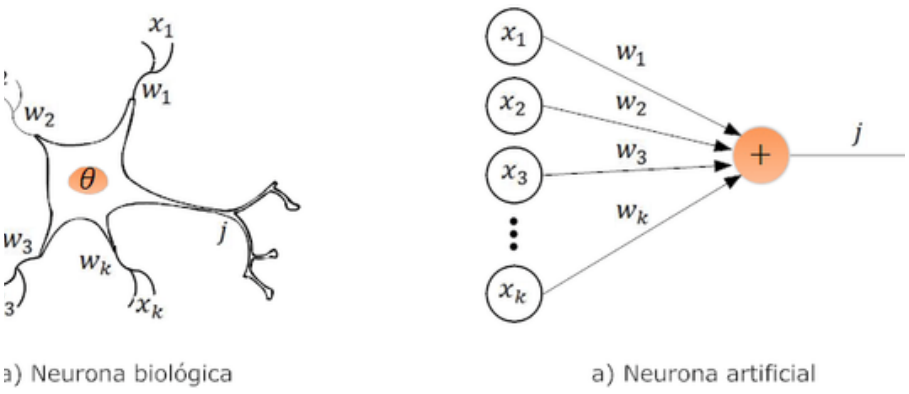
# LOS ORÍGENES DE LA INTELIGENCIA ARTIFICIAL

Aunque en los últimos años la inteligencia artificial parece estar en boca de todos, especialmente gracias a avances como ChatGPT o los generadores de imágenes, es importante destacar que **no se trata de una tecnología nueva ni improvisada.** La IA es, en realidad, **una disciplina con casi un siglo de historia, cuyas raíces se remontan a la década de 1940.**

A lo largo de su desarrollo, **encontramos una línea de tiempo marcada por avances y retrocesos**, períodos de entusiasmo (las llamadas edades doradas) y momentos de desencanto (conocidos como los inviernos de la IA). Este contexto histórico permite comprender que **la inteligencia artificial no es un fenómeno repentino, sino el resultado de un proceso colectivo, científico y tecnológico, que ha requerido décadas de maduración para llegar al punto actual, en el que se cruza con investigaciones en múltiples áreas**.

## La neurona artificial: el modelo fundacional

No es un fenómeno repentino, sino el resultado de un proceso colectivo, científico y tecnológico, que ha requerido décadas de maduración.




Optimización de la Estimación de DOA en Sistemas de Antenas Inteligentes usando criterios de Redes Neuronales - Scientific Figure on ResearchGate. Disponible acá. Consultado 08/2025.

El modelo matemático en el que aún hoy se sustenta la inteligencia artificial es el de la **neurona artificial**. Surgido en los años cuarenta, **está inspirado en la neurona biológica y funciona de manera análoga.**

La idea es sencilla: cada entrada (X) representa un estímulo —por ejemplo, un dato o una señal—, y cada uno de esos estímulos se pondera con un peso (W) que refleja su importancia relativa. El producto de cada entrada por su peso se suma, y el resultado se conoce como nivel de activación, denotado por la letra $n$. Ese nivel de activación pasa luego por una función (la función de activación $f(n)$) que determina si la neurona produce o no una respuesta. Esta respuesta se denomina salida y se representa como $y$.

Este modelo simple (una suma ponderada seguida de una función no lineal) constituye el ladrillo fundamental de todas las redes neuronales. Cualquier red, por más compleja que sea (desde un clasificador de proteínas hasta ChatGPT), está construida con millones o miles de millones de estas unidades. Lo que cambia es la cantidad de neuronas, la forma en que se conectan y cómo se ajustan los pesos.

# El taller de Dartmouth: nacimiento de la disciplina

Infobae. (s.f.). Participantes del Taller de Dartmouth (1956), considerado el inicio oficial de la inteligencia artificial. [1]

Acuñó el término inteligencia artificial, sino que se considera el acta de nacimiento simbólica de la IA como disciplina científica formal.

Contar con un modelo no basta para que surja un campo de investigación. En un contexto de fuerte optimismo tecnológico, después de la Segunda Guerra Mundial, con computadoras cada vez más potentes y una fe casi ilimitada en el progreso científico, se organizó un evento que quedaría en la historia: **el taller de verano en Dartmouth College, en 1956.**

La propuesta era ambiciosa, **reunir durante dos meses a especialistas de distintas disciplinas** —matemáticos, ingenieros, físicos, filósofos— para reflexionar en conjunto sobre **cómo lograr que las máquinas puedan utilizar el lenguaje, razonar, resolver problemas y mejorar por sí mismas.**

Este taller no solo acuñó el término inteligencia artificial, sino que se considera el acta de nacimiento simbólica de la IA como disciplina científica formal. A partir de entonces comenzaron a formarse grupos de investigación, conferencias, mecanismos de financiamiento y, sobre todo, una comunidad científica con objetivos compartidos.

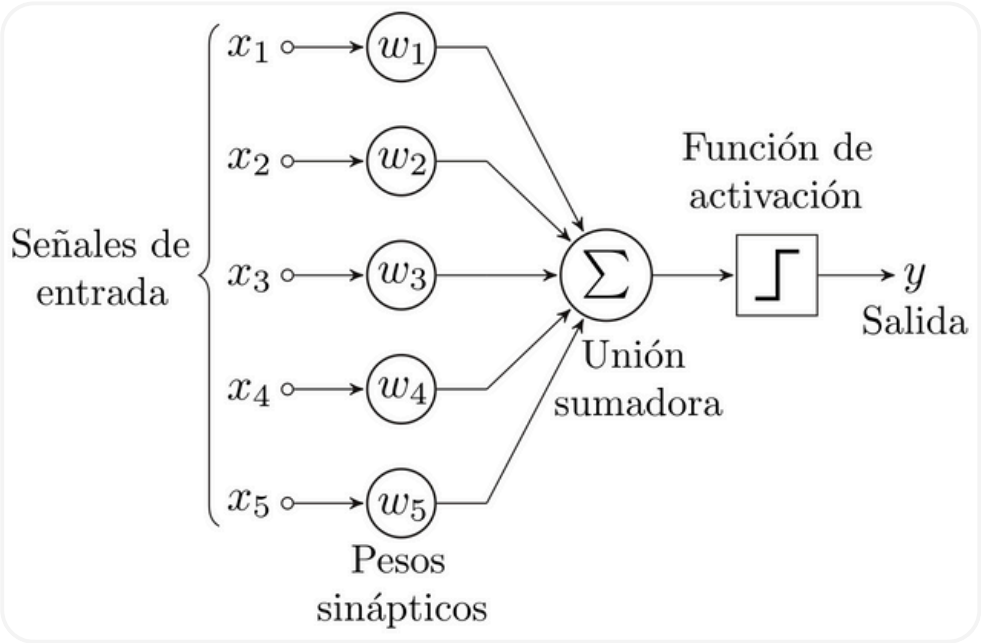
De esta manera, se estableció el **concepto fundacional de inteligencia artificial: técnicas que permiten que una computadora emule ciertos aspectos del comportamiento humano.**

# El perceptrón: la máquina que aprende

Dos años después del taller de Dartmouth, lo que hasta ese momento era solo un modelo matemático dio un salto sorprendente: **una neurona artificial ahora podía aprender**. Este avance fue tan disruptivo que algunos medios titularon: ¡La máquina que piensa!.

Se trataba del **perceptrón simple**, desarrollado por **Frank Rosenblatt** en **1958**. ¿Cuál era su innovación? El perceptrón no se limitaba a aplicar una función sobre los datos, sino que ajustaba sus pesos a partir de ejemplos. Este mecanismo es lo que hoy conocemos como **aprendizaje supervisado**.

Fue un momento fundacional porque, por primera vez, una máquina no solo procesaba datos, sino que aprendía de ellos.




Wikipedia contributors. (s.f.). Artificial neuron – estructura básica. En Wikipedia. Recuperado el 07/2025, de Infobae. (s.f.).

La fórmula correspondiente muestra cómo se ajusta cada peso: se parte del valor actual del peso y se le suma una corrección que depende del error cometido (la diferencia entre lo predicho y lo que debía predecirse), multiplicada por la tasa de aprendizaje. Este fue **un momento fundacional porque, por primera vez, una máquina no solo procesaba datos, sino que aprendía de ellos**. Aunque se trataba de una red de una sola capa, muy limitada, sentó las bases de todo el aprendizaje automático.

## Despleguemos un ejemplo

Supongamos que tenemos que enseñarle a una computadora a distinguir entre dos tipos de proteínas: aquellas que forman parte de las mitocondrias y las que no.

Las mitocondrias son estructuras dentro de nuestras células que funcionan como pequeñas centrales de energía. Algunas proteínas trabajan allí, y otras no. Entonces, si logramos entrenar un modelo que reciba información de una proteína y pueda decirnos si pertenece o no a una mitocondria, habremos resuelto un problema de clasificación binaria: solo hay dos posibles respuestas.

En el lenguaje de la computadora, esas dos respuestas se representan como:

- 1: la proteína es mitocondrial.
- 0: la proteína no es mitocondrial.

Este es un ejemplo didáctico: no vamos a trabajar con proteínas reales, sino con datos simplificados para mostrar cómo aprende el modelo.

## Paso a paso: de los datos a la predicción

1

**Convertir la información en números**
Para que una computadora pueda trabajar, nesitamos transformar las características de cada proteína (por ejemplo, su tamaño, su peso molecular o ciertas señales químicas) en una lista de números. A esa lista la llamamos vector de características.

2

**Empezar desde cero**
Al principio, la red neuronal no sabe nada. Sus parámetros internos (llamados pesos y sesgo) se inicializan con valores aleatorios.

3

**Hacer una primera predicción**
La red combina los números de la proteína usando una suma ponderada (cada dato se multiplica por un peso) y obtiene un valor, que en este caso dio −3,055. Ese valor pasa por una regla muy simple: si es mayor o igual a cero, predice 1 (sí, es mitocondrial). Si es menor a cero, predice 0 (no lo es). Como en este caso dio negativo, la red predijo que la proteína no era mitocondrial.

A partir de datos convertidos en números, la red hace una predicción inicial, compara con la respuesta correcta y ajusta sus parámetros.

4

**Comparar con la respuesta correcta**
Sabemos de antemano cuál era la categoría verdadera de esa proteína (porque está en nuestro conjunto de entrenamiento). En este ejemplo, sí era mitocondrial. La predicción fue incorrecta.

5

**Ajustar y volver a intentar**
Cada vez que la red se equivoca, usa esa diferencia para ajustar sus pesos y su sesgo. Así, poco a poco va aprendiendo a mejorar. Luego pasa al siguiente ejemplo y repite el proceso.

**Cuando la red recorre todo el conjunto de ejemplos una vez, decimos que completó una época. Con muchas épocas, la red va corrigiéndose y mejora su capacidad para distinguir entre las dos clases.**

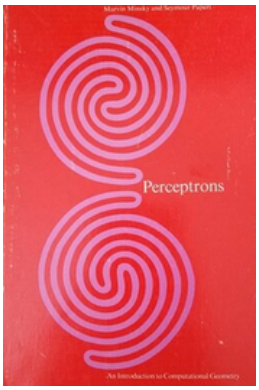
## Primer límite: el problema de la separabilidad

El perceptrón simple solo puede separar datos con líneas rectas. El problema XOR mostró que hay casos donde eso no alcanza y marcó el inicio del primer invierno de la inteligencia artificial.

Todo parecía indicar que la inteligencia artificial avanzaba sin freno. Las primeras máquinas que aprendían habían despertado la imaginación del mundo científico. Sin embargo, no pasó mucho tiempo hasta que se chocaron con un límite que resultó decisivo.

El **perceptrón simple**, que tanto entusiasmo había generado, **demostró ser incapaz de resolver ciertos tipos de problemas**. Su manera de aprender era muy limitada: solo podía trazar una línea (o un plano, si los datos tenían más dimensiones) para separar los ejemplos de una clase de los de la otra. Esto funciona solo cuando los datos están bien ordenados, cuando son linealmente separables.

## El impacto de esta limitación




La publicación de este resultado en 1969, en el libro Perceptrons de Marvin Minsky y Seymour Papert, fue un verdadero balde de agua fría para la comunidad científica. El entusiasmo se transformó en desencanto: muchos investigadores abandonaron el campo, se frenaron los fondos y comenzó lo que hoy se conoce como el primer invierno de la inteligencia artificial.

Wikipedia contributors. (s.f.). Portada del libro Perceptrons (1969). En Wikipedia. Recuperado el 07/25, de https://en.wikipedia.org/wiki/Perceptrons

# DEL APRENDIZAJE AUTOMÁTICO AL SALTO HACIA LAS REDES PROFUNDAS


Imagen de la película AlphaGo (2017)

## La idea que cambió todo: capas ocultas

Sin embargo, la historia no terminó ahí. Aunque el entusiasmo se había enfriado, la investigación siguió. Y en ese contexto alguien propuso una idea simple y a la vez brillante:

**¿Qué pasaría si, entre la entrada y la salida, agregamos una capa oculta de neuronas?**

Agregar capas ocultas permitió que las redes aprendan relaciones complejas y resuelvan problemas antes imposibles.

No cualquier neurona, sino **neuronas con funciones de activación no lineales y diferenciables, capaces de transformar los datos de maneras mucho más ricas.**

Ese pequeño cambio abrió un mundo nuevo, ya no estábamos limitados a trazar líneas rectas: ahora el modelo podía aprender fronteras de decisión curvas, complejas, y con eso… ¡resolver el problema XOR!

Así nació el concepto de red neuronal, ya no hablamos de una sola neurona, sino de muchas, organizadas en capas y conectadas entre sí. **Cada neurona tiene su propia función de activación; cada conexión, su peso.** Y si uno lo piensa un segundo esto empieza a parecerse bastante más al cerebro.

## El siguiente desafío: cómo enseñarles a todas las capas

Con esta arquitectura más compleja se abrió una nueva pregunta muy importante ¿cómo hacemos para entrenar una red que tiene varias capas?

Hasta ese momento, **el aprendizaje se aplicaba solo a la capa de salida. Pero ahora había que ajustar los pesos de todas las capas, incluso de aquellas que no están directamente conectadas con la salida. ¿Cómo corregimos algo que está escondido en el medio?**

La respuesta fue otro descubrimiento fundamental: el algoritmo de retropropagación del error (backpropagation). Su idea es elegantísima:

- Primero, la red hace su predicción y calculamos cuánto se equivocó (el error).
- Después, ese error se propaga hacia atrás, capa por capa, calculando cuánto aportó cada peso a ese error.
- Con esa información, se ajustan todos los pesos de la red, no solo los de la última capa.

## Cómo se diseña una red: la idea de arquitectura

Hasta ahora hablamos de redes neuronales en general, pero queda una pregunta importante: **¿cómo se diseña una red? ¿Qué decisiones hay que tomar para armarla?**

En este punto entra en juego el concepto de **arquitectura de red**.

Cuando hablamos de arquitectura nos referimos a tres aspectos centrales:

- **Conectividad:** cómo están unidas las neuronas entre sí. En la forma más simple, cada neurona de una capa se conecta con todas las de la siguiente. A esto se lo llama una red densa o fully connected.
- **Profundidad:** cuántas capas ocultas tiene la red. Cuantas más capas, mayor capacidad para aprender representaciones complejas. De ahí viene el término *deep learning*.
- **Ancho:** cuántas neuronas tiene cada capa. También este número afecta cuánto puede aprender el modelo.

La arquitectura (su conectividad, profundidad y ancho) define qué tan bien puede aprender, cuánto tarda en entrenar y cuán capaz es de generalizar.

En resumen, **al diseñar una red tenemos que decidir cuán profunda, cuán ancha y cuán conectada va a ser**. Cada una de estas decisiones tiene consecuencias en el rendimiento, en el tiempo de entrenamiento y en la capacidad de la red para generalizar. Ajustar esta arquitectura al problema que queremos resolver es parte central del trabajo.

## La diversificación de arquitecturas

Con el tiempo quedó claro que la arquitectura clásica, **totalmente conectada, no era siempre la mejor**. Sirve para algunos problemas, pero no para todos. Y así comenzó un proceso de diversificación: los investigadores empezaron a diseñar arquitecturas especializadas, adaptadas al tipo de dato.

Por ejemplo:

- **Cuando trabajamos con imágenes**, los datos tienen una estructura espacial muy rica: hay píxeles ordenados en alto y ancho, bordes, texturas y patrones locales. Para aprovechar esa estructura surgieron las redes convolucionales o **CNN**, que usan filtros que se mueven por la imagen extrayendo características relevantes. Estas redes transformaron por completo el reconocimiento de imágenes, desde diagnóstico médico hasta visión computacional.

- **Cuando trabajamos con secuencias**, como texto, ADN, audio o series temporales, necesitamos redes que tengan en cuenta el orden y el contexto. Para eso se desarrollaron **las redes recurrentes o RNN**, que **procesan los datos paso a paso y mantienen una especie de "memoria" de lo que pasó antes**. Estas redes fueron clave en tareas como la traducción automática, el análisis de texto y algunas áreas de la bioinformática.

En pocas palabras, la inteligencia artificial aprendió a adaptar la forma de sus redes al tipo de dato con el que trabaja, y eso abrió un abanico enorme de posibilidades.

## Las RNN y sus límites

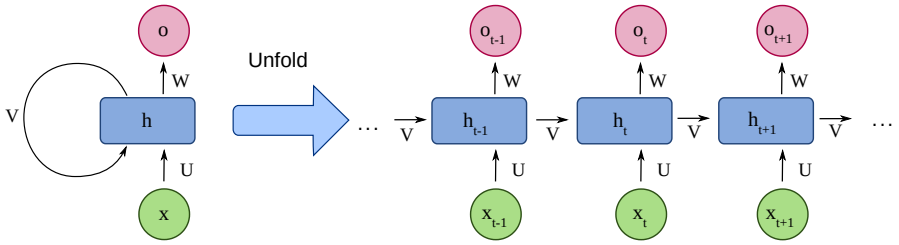
Las redes recurrentes representaron un avance enorme: por primera vez era posible procesar datos secuenciales considerando lo que venía antes. Pero con el tiempo aparecieron límites importantes.

La arquitectura (su conectividad, profundidad y ancho) define qué tan bien puede aprender, cuánto tarda en entrenar y cuán capaz es de generalizar.

Por un lado, el famoso problema del **desvanecimiento del gradiente**: cuando la secuencia es muy larga, el error que se propaga hacia atrás se va diluyendo. En la práctica esto significa que **las RNN tienen "memoria corta": les cuesta aprender relaciones a largo plazo**.

Por otro lado, al procesar la información de manera estrictamente secuencial, las RNN no pueden aprovechar el cálculo en paralelo. **Cada palabra depende de la anterior, así que todo debe hacerse paso a paso. Esto las vuelve lentas e ineficientes cuando hay grandes volúmenes de datos.**

Estas limitaciones impulsaron la búsqueda de una nueva arquitectura que pudiera superar esos cuellos de botella. Y así nació el Transformer.




Red neuronal recurrente básica comprimida (izquierda) y desplegada (derecha)

By fdeloche - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=60109157

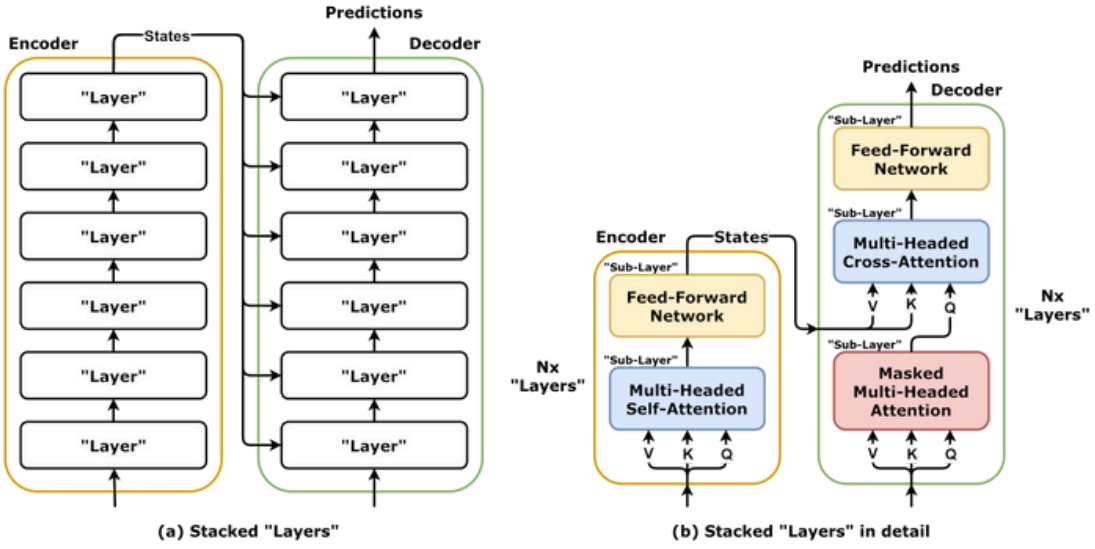
## El Transformer: una nueva forma de procesar secuencias

En 2017 se publicó un artículo que cambió el rumbo de la inteligencia artificial: Attention is All You Need. Allí se presentó el **Transformer**, una arquitectura completamente nueva para procesar secuencias.

La idea rompe con la secuencia paso a paso de las RNN. El Transformer **procesa todas las palabras al mismo tiempo, en paralelo, aprovechando al máximo los recursos de cómputo modernos**.

¿Cómo lo logra? Gracias a un mecanismo llamado **self-attention**, que permite que **cada palabra "mire" a todas las demás dentro de la oración y calcule automáticamente cuáles son más importantes**. Así, cada palabra se representa no solo por sí misma sino también en función de su contexto. Este mecanismo le da al modelo un contexto mucho más amplio y preciso que el de las RNN.

El resultado es una arquitectura capaz de **aprender dependencias a largo plazo de forma más rápida y eficiente**. Esta arquitectura es hoy la base de los modelos modernos que usamos en ciencia, medicina, arte y lenguaje.




Un transformador se compone de capas de codificador y capas de decodificador apiladas.

## Cómo entienden el lenguaje los Transformers

Para que una red pueda trabajar con texto, lo primero es **convertir las palabras en números**. Ese proceso se llama **vectorización**: a cada palabra se le asigna un vector de números que refleja ciertas propiedades estadísticas. Estos vectores se llaman **embeddings.**

Los embeddings se entrenan junto con el modelo, de modo que palabras con significados parecidos terminan cerca unas de otras en este espacio de vectores. Por ejemplo, king y queen o man y woman. Sin embargo, los embeddings clásicos tienen una limitación: **son estáticos**. La representación de una palabra no cambia aunque cambie la oración. Así, bank se vectoriza igual si hablamos de un banco de dinero o de un banco de peces.

El mecanismo de **self-attention** resuelve este problema. Gracias a él, cada palabra puede "mirar" a las demás en la frase y ajustar su significado según el contexto. Es como si cada palabra se preguntara: "¿a quién debo prestar atención para entender lo que significo en esta oración?".

El embedding convierte palabras en vectores que guardan relaciones semánticas, mientras que el self-attention permite ajustar su significado según el contexto.

## ¿Qué hace en realidad un modelo basado en Transformers?

Con todo esto entendido, podemos resumir la tarea principal de un Transformer en una sola frase: predecir la siguiente palabra más probable, dado un contexto.

El modelo toma las palabras que ya tiene, calcula —en base a todo lo que aprendió— cuál sería la palabra más coherente, más probable, más natural… y la escribe. Luego repite el proceso con la palabra nueva agregada al contexto una y otra vez.

Por eso los modelos actuales pueden responder preguntas, redactar textos, traducir, resumir artículos científicos o incluso generar código. Son modelos probabilísticos del lenguaje, entrenados a una escala gigantesca, que aprendieron a anticipar el flujo del lenguaje humano con una precisión sorprendente.

# Bibliografía


Kim, J. (2025). Artificial Intelligence and Clinical Practice. Journal of Korean Medical Science, 40, e187. https://doi.org/10.3346/jkms.2025.40.e187

Mallapaty, S. (2025, May 16). AI tools are transforming science — here's how researchers can stay on top. Nature. https://www.nature.com/articles/d41586-025-01463-8.pdf

Wiley. (2025, February). ExplanAItions: The future of AI in research and publishing. https://www.wiley.com/content/dam/wiley-dotcom/en/b2c/content-fragments/explanaitions-ai-report/pdfs/Wiley_ExplanAItions_AI_Study_February_2025vers1.pdf

Nature Portfolio. (n.d.). Guidance on the use of generative AI in Nature Portfolio journals. Nature. https://www.nature.com/nature-portfolio/editorial-policies/ai

Nature Portfolio. (n.d.). Editorial policies on the use of AI tools. Nature. https://www.nature.com/nature-portfolio/editorial-policies/ai

Bloghemia. (2025, April). La estupidez insuficiente de la IA. https://www.bloghemia.com/2025/04/la-estupidez-insuficiente-de-la-ia-por.html

Luger, G. F. (2025). Artificial Intelligence: Principles and Practice. Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-031-57437-5