

# Tarea 2 - Arquitecturas de Computación

David Santiago López Maldonado

Agosto de 2025

## Resumen

Por medio del siguiente documento se muestran diversas arquitecturas de computación que presentan un cambio importante frente a la computación clásica que aborda mayormente una arquitectura Von Neumann. Teniendo cinco (5) tipos de arquitectura diferentes como lo son: *Computación cuántica*, *Computador neuromórfico*, *Ordenador biológico*, *Computación Heterogénea* y *Computación de borde*.

Cada uno de estos modelos presenta fundamentos, arquitecturas y aplicaciones diferentes, pero todos comparten el objetivo común de mejorar el rendimiento, la eficiencia y la capacidad de procesamiento de la información frente a los desafíos actuales de la tecnología. Asimismo, se explorarán sus ventajas, desventajas, principales hitos históricos y el estado actual de desarrollo, con el fin de comprender su potencial y limitaciones.

## 1. Computación cuántica

La computación cuántica aplica los conceptos físicos y matemáticos de la mecánica cuántica a la informática para poder procesar información de una forma mucho más rápida y eficiente que la computación clásica.

En lugar de trabajar con bits que solo representan un estado o un nivel (0 o 1), la computación cuántica emplea qubits, es decir, bits cuánticos que gracias al estado de superposición pueden representar ambos estados o niveles (0 y 1) al mismo tiempo. Esta propiedad junto con algunas otras, permiten explorar y calcular varias soluciones o respuestas paralelamente. Todo esto como alternativa a problemas que los computadores tradicionales o supercomputadores no pueden resolver o tardan demasiado tiempo en resolverlos [2][3].

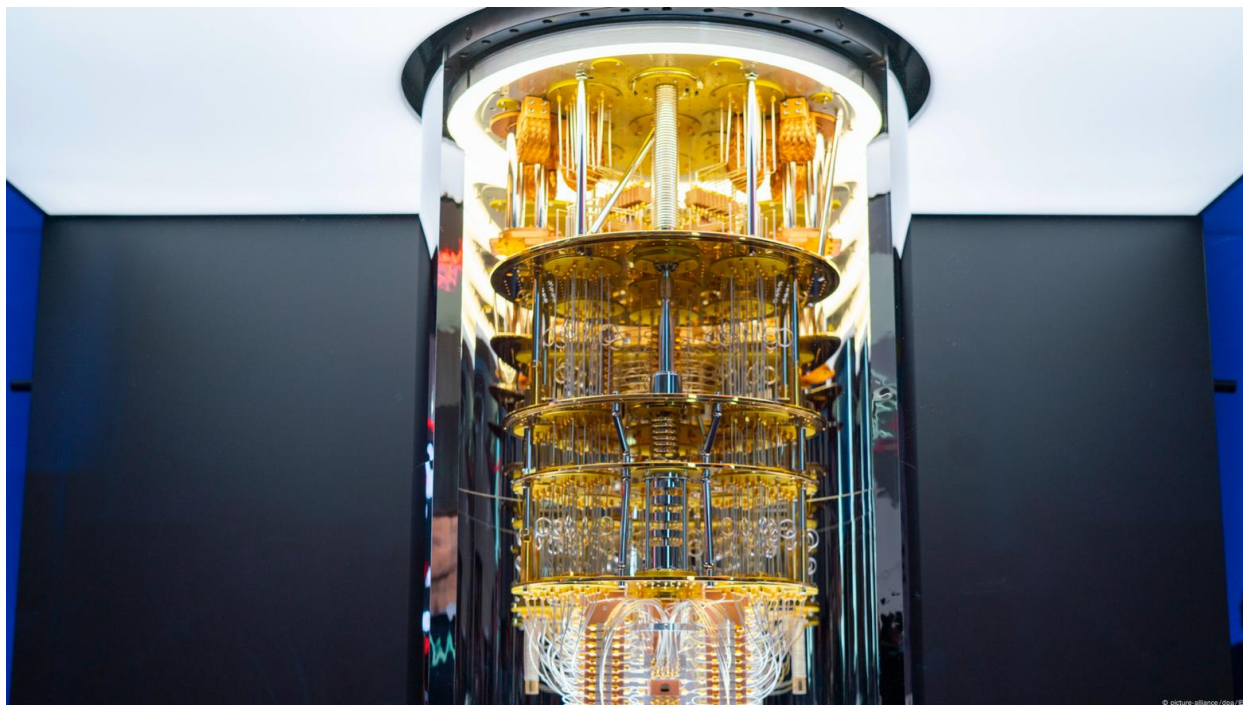


Figura 1: Computadora cuántica IBM.

## 1.1. Arquitectura de una computadora cuántica

### ■ Unidad de Procesamiento Cuántico (QPU):

Esta unidad de procesamiento cuántico es análoga a la CPU de un computador tradicional, sin embargo, como se explicó antes, no se procesan bits comunes sino bits cuánticos (qubits) los cuales funcionan mediante fenómenos cuánticos como la superposición y el entrelazamiento. Estos procesadores son los encargados de realizar las operaciones cuánticas que se definieron mediante algoritmos cuánticos, incluyendo las compuertas cuánticas, que son las que alteran los estados de los qubits. Es también conocido como el corazón de la computadora, alberga los qubits y controla las operaciones que la máquina hará. Los componentes de la QPU son los siguientes:

- Qubits : Son las unidades fundamentales de la computación cuántica. Son capaces de representar dos estados simultáneamente (0 y 1) mediante el fenómeno de superposición. Los qubits se pueden crear mediante tecnologías como:
  - circuitos superconductores : Son circuitos fabricados con materiales superconductores que están enfriados con una temperatura cercana al cero absoluto ( $0^{\circ} \text{K} = -273.15^{\circ} \text{C} = -459.63^{\circ} \text{F}$ ).
  - Iones atrapados : Son átomos cargados suspendidos en trampas electromagnéticas.
  - Fotones : Aprovechan partículas de luz para procesar información.
  - Puntos cuánticos : Son nanoestructuras que aíslan electrones.
  - Qubits de Espín : Dependen del espín de los electrones o núcleos.
- Compuertas cuánticas : Análogas a las compuertas lógicas, las compuertas cuánticas manipulan qubits de formas controladas para poder realizar cálculos complejos y resolver problemas que serían imposibles de resolver con computadores convencionales.
- Interconexiones : Permiten la comunicación entre qubits y demás QPU para poder escalar los computadores cuánticos y realizar cálculos más complejos

- **Electrónica de control:** Gestiona las operaciones del computador cuántico, genera y envía señales precisas (microondas, radiofrecuencia o láser) que manipulan los qubits y las compuertas cuánticas en la QPU, además sincronizan las operaciones para que los resultados precisos sean garantizados.
- **Infraestructura de refrigeración:** Para garantizar la coherencia de los qubits y minimizar el ruido térmico, es necesario que el computador se mantenga a temperaturas cercanas al cero absoluto. Este sistema está alimentado por refrigeradores de dilución que mantienen el hardware a temperaturas de unos 10mK-20mK y permiten el aislamiento del procesador al calor y al ruido. Estos refrigeradores funcionan mediante la mezcla de isótopos de Helio. Los aisladores crio génicos protegen la QPU de interferencias térmicas y del ruido externo. Los amplificadores crio génicos amplifican las señales débiles emitidas por los qubits para medirlas con precisión sin agregar ruido adicional.
- **Infraestructura de corrección de errores cuánticos:** Conocida como QEC (Quantum Error Correction), esta infraestructura se encarga de darle fiabilidad a los qubits, estos son demasiado sensibles a interferencias ambientales y a imperfecciones de la máquina. Los estados cuánticos de los qubits pueden degradarse rápidamente y esto conlleva a errores computacionales y entre mayor escalabilidad, mayor probabilidad de error, de ahí la justificación de esta infraestructura. Para evitar la alteración en los estados cuánticos de los qubits, los qubits lógicos se codifican en diferentes qubits físicos, permitiendo la detección de errores. Los circuitos especializados detectan la ubicación de los errores y se aplica la corrección correspondiente como, por ejemplo, invertir los qubits para poder restaurar el qubit lógico.
- **Interfaz de computación clásica:** Esta interfaz de computación clásica controla el flujo de operaciones de la computadora cuántica, sirve como un puente entre lo convencional y lo cuántico. Es capaz de gestionar la entrada y salida de los cálculos cuánticos, asiste en las tareas como los cálculos de corrección de errores e interpreta los resultados de las mediciones obtenidas.
- **Interconexiones cuánticas:** Permiten las comunicaciones cuánticas y las redes cuánticas, son las responsables de transmitir información cuántica entre computadoras cuánticas. Son importantes para la escalabilidad de estas computadoras y para permitir el procesamiento cuántico colaborativo en diferentes sistemas.

## 1.2. Historia

- **1900 - 1930**
  - 1900: Max Planck descubrió que la energía es emitida y absorbida por paquetes discretos llamados "cuanta", esto solucionó el problema de radiación de cuerpo negro que los físicos aún no podían entender. Además, desafió la física clásica al introducir la discretización a escala atómica, dando como origen la teoría cuántica.
  - 1913: Niels Bohr desarrolló un modelo atómico en donde postulaba que los electrones se movían en niveles específicos y cuantificados de energía. Explica el por qué los átomos, en longitudes de onda específicas absorben y emiten luz (Fotones con cantidades precisas de energía).
  - 1925-1926: La mecánica matricial fue desarrollada por Werner Heisenberg, la mecánica ondulatoria formulada por Erwin Schrödinger. Por medio de estos marcos matemáticos se estableció la mecánica cuántica moderna, que establece que las partículas están en una dualidad de onda-partícula, siguiendo el principio de incertidumbre y sentando las bases para la computación cuántica.
- **1930 - 1940**
  - 1935: Einstein, Podolsky y Rosen publicaron un experimento mental en el que se destacó el fenómeno de entrelazamiento cuántico, recurso que se volvió fundamental en la computación cuántica.
  - 1939: Isidor Rabi mostró cómo pueden cambiar los estados cuánticos de un núcleo atómico mediante ondas de radio por medio de la Resonancia Magnética Nuclear. Esta técnica fue la primera en poder controlar los estados cuánticos mediante radiación electromagnética, estableciendo los

principios que evolucionaron en métodos capaces de manipular los qubits empleados en las computadoras cuánticas.

■ 1950 - 1970

- 1952: Félix Bloch y Edward Mills Purcell desarrollaron métodos precisos que permitieron a los científicos observar precisamente las propiedades cuánticas de los átomos. Esto permitió establecer herramientas experimentales esenciales para el estudio y manipulación de los sistemas cuánticos.
- 1965: El teorema de Bell proporcionó el marco para eventuales experimentos que confirmarían la naturaleza no local del entrelazamiento y cuestionarían nuestra comprensión fundamental de la realidad.
- 1972: Alain Aspect, a inicios de los años 80, realizó experimentos que confirmaron el entrelazamiento cuántico, validando la acción fantasmal a distancia propuesta por la mecánica cuántica. Sus hallazgos sentaron bases fundamentales para el desarrollo de la computación cuántica.

■ 1980 - 1990

- 1981: Richard Feynman propuso que solo las computadoras cuánticas podían simular eficazmente sistemas cuánticos, planteando que resolverían ciertos problemas mucho más rápido que las clásicas y estableciendo las bases teóricas de la computación cuántica.
- 1983: Alain Aspect confirmó experimentalmente las violaciones de la desigualdad de Bell, demostrando de forma definitiva la realidad del entrelazamiento cuántico y consolidando su papel como base para la computación cuántica.
- 1985: David Deutsch formuló el primer modelo teórico de una computadora cuántica universal, mostrando que las puertas cuánticas podían realizar cualquier cálculo cuántico. Con ello extendió la computación universal al ámbito cuántico y planteó la posibilidad de resolver problemas imposibles para las computadoras clásicas.
- 1994: Peter Shor creó un algoritmo de factorización cuántica que resolvía grandes números mucho más rápido que los métodos clásicos, demostrando la ventaja práctica de la computación cuántica y generando gran impacto en la criptografía y en la inversión en este campo.
- 1995: El NIST demostró experimentalmente la primera puerta cuántica controlada-NOT (CNOT), validando la posibilidad física de operaciones básicas entre cúbits y marcando el paso de la teoría a la implementación práctica de circuitos cuánticos.

■ 2000 - Presente

- 2001: Georges-Olivier Reymond y otros científicos demostraron el uso de pinzas ópticas para atrapar átomos neutros individuales, logrando un control cuántico preciso partícula por partícula. Este avance abrió una nueva plataforma para el procesamiento de información cuántica y dio origen a la tecnología de Pasqal.
- 2011: D-Wave One (128 qubits) se convierte en la primera computadora cuántica disponible comercialmente
- 2016: IBM ofrece acceso público a computación cuántica vía su plataforma en la nube (IBM Quantum Experience), democratizando el acceso a hardware cuántico experimental.
- 2016 - 2019: Google, en colaboración con NASA, anuncia la consecución de la supremacía cuántica con su procesador de 54 qubits, al ejecutar una tarea que afirmaban sería impracticable para computadoras clásicas. IBM refutó parte de esta afirmación, generando debate científico.
- 2018 - 2025: Esta fase se conoce como la era NISQ (Noisy Intermediate-Scale Quantum), caracterizada por dispositivos con hasta 1000 qubits, aunque aún sin corrección de errores completa. John Preskill acuña el término en 2018, y en 2023 se supera la barrera de los 1,000 qubits físicos (Atom Computing, 1,180 qubits)

### 1.3. Ventajas y Desventajas

Ventajas	Desventajas
Velocidad y eficiencia en ciertos problemas: resolución exponencialmente más rápida que en computadores clásicos.	Fragilidad de los cúbits: muy sensibles al ruido y la decoherencia.
Simulación precisa de sistemas cuánticos: moléculas, materiales y reacciones químicas.	Necesidad de condiciones extremas: refrigeración casi al cero absoluto.
Optimización avanzada en logística, finanzas, energía y telecomunicaciones.	Escalabilidad limitada: actualmente solo decenas o cientos de cúbits.
Criptografía cuántica: desarrollo de comunicaciones ultra seguras (QKD).	Altos costos y complejidad técnica en infraestructura y mantenimiento.
Posibilidad de resolver problemas inalcanzables para supercomputadores clásicos.	Riesgo para la seguridad actual: amenaza a los sistemas criptográficos clásicos (RSA, ECC).
	Madurez tecnológica insuficiente: aún en fase experimental con aplicaciones limitadas.

### 1.4. Conceptos clave

- **Superposición:**  
Habla de la capacidad de una partícula de estar en más de un estado al mismo tiempo. Imaginemos una moneda lanzada al aire: mientras está girando, no es solo cara o sello, sino una mezcla de ambos. Cuando cae al piso (cuando se mide), se convierte en un resultado concreto.
- **Entrelazamiento:**  
Establece que, sin importar la distancia, dos partículas cuánticas pueden conectarse de tal forma que lo que le sucede a una, instantáneamente afecta a la otra. Si entrelazamos dos monedas cuánticas y una resulta "cara", automáticamente la otra será "sello", incluso si están separadas por kilómetros.
- **Interferencia cuántica:**  
Al estar las partículas en el estado de superposición, los posibles estados pueden sumarse o anularse, esto permite crear algoritmos que favorezcan respuestas o resultados correctos. Pensemos en dos olas de agua: si coinciden, se refuerzan (interferencia constructiva); si están opuestas, se cancelan (interferencia destructiva).
- **Medición probabilística:**  
Al tomar la medición de un qubit en estado de superposición, su medición no es concreta sino se determina por probabilidad. Esto explica el por qué es necesario repetir muchas veces los algoritmos cuánticos para obtener un resultado confiable. Un qubit puede estar en 70 por ciento de probabilidad de ser 0 y 30 por ciento de ser 1. Al medirlo, el resultado será 0 la mayoría de veces, pero a veces será 1.
- **Desafío de decoherencia:**  
Sucede cuando un qubit pierde su estado cuántico debido a interacciones con el ambiente, ya sea ruido, calor o radiación. Es como si la moneda en el aire fuera golpeada por el viento antes de caer: ya no refleja la verdadera probabilidad, sino un error.
- **Tipos de comunicación cuántica**
  - **Distribución cuántica de claves :** Esta distribución usa entrelazamiento para generar claves criptográficas imposibles de detectar.
  - **Tele transportación cuántica :** transmite el estado cuántico de una partícula a otra distante, usando entrelazamiento, aunque la partícula en sí no viaja.
  - **Redes cuántica :** Es la idea de un "Internet cuántico" basado en cúbits entrelazados.

- Compuertas cuánticas:

Análogas a las compuertas lógicas clásicas, las compuertas cuánticas manipulan los qubits según un algoritmo o circuito.

- X gate (NOT cuántico) : invierte el estado del qubit.
- H gate (Hadamard) : Coloca a los qubits en estado de superposición.
- CNOT (Controlled-NOT) : genera entrelazamiento, cambia el segundo cúbit solo si el primero es 1.

Si tenemos dos cúbits: El primero = 1, El segundo = 0. Después de pasar por una CNOT el segundo se convierte en 1.

## 2. Computador Neuromórfico

Un computador neuromórfico (también llamado ingeniería neuromórfica) es un sistema de hardware y software diseñado para imitar el funcionamiento del cerebro humano. Integra elementos inspirados en neuronas y sinapsis reales, y procesa información de forma distribuida, paralela y altamente eficiente. Este enfoque busca superar limitaciones de la arquitectura tradicional de von Neumann, especialmente en términos de consumo de energía y latencia.

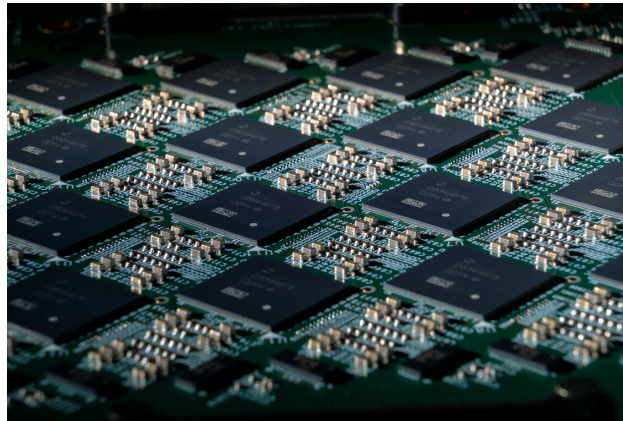


Figura 2: Chips neuromórficos de 5ta Generación.

### 2.1. Arquitectura de un computador neuromórfico

- Entrada y codificación sensorial.

Son módulos que toman señales externas como imagen, audio, etc, y las convierten en eventos (o spikes) en lugar de cuadros continuos de datos. Esto reduce datos y latencia, solo emite algo cuando cambia la escena. Por ejemplo, cámaras de eventos (DVS) generan un evento por pixel cuando detectan un cambio de luminancia; si nada cambia, no emiten casi nada.

- Núcleos neurosinápticos.

Agrupar las neuronas y sus sinapsis. Cada núcleo recibe spikes en axones (entradas), los pondera con una matriz sináptica y actualiza el estado de cada neurona; cuando una neurona supera umbral, emite un spike a la red.

- Motor de plasticidad.

Es la lógica cerca de las sinapsis que actualiza pesos con reglas locales (por ejemplo, STDP, refuerzo, aprendizaje hebbiano). Permite aprendizaje on-chip sin ir a CPU externa, manteniendo el procesamiento en memoria y event-driven

- **Memoria sináptica y representación de pesos.**  
Es dónde viven los pesos y metadatos (retrasos, “tipo de axón”, etc.). Suele ser SRAM distribuida por núcleo; en plataformas analógicas/memristivas puede ser NVM.dónde viven los pesos y metadatos (retrasos, “tipo de axón”, etc.). Suele ser SRAM distribuida por núcleo; en plataformas analógicas/memristivas puede ser NVM.Habilita el cómputo en memoria: el acceso a los pesos ocurre localmente para cada spike, sin tráfico a DRAM externo.
- **Red de interconexión.**  
Es la “mallla” que transporta spikes como paquetes ligeros con la dirección del emisor o del destino: Address-Event Representation (AER). Enruta spikes de forma asíncrona y multicast (un spike se copia a muchos destinos), escalando a redes grandes con baja latencia.
- **Clúster neuromórfico.**  
Son topologías multi-chip/multi-nodo con routers de alto fan-out y encaminamiento tolerante a fallos. Permite simular/controlar redes neuronales masivas en tiempo biológico.
- **Temporización y ejecución asíncrona.**  
Muchos diseños evitan un reloj global; las actualizaciones ocurren al arribo de eventos o por clocks locales. Esto reduce consumo (no hay conmutación si no hay eventos) y elimina cuellos de botella de sincronización global.
- **Control y monitorización.**  
Son microcontroladores/cores de servicio que cargan configuraciones, inician experimentos, coleccionan métricas y comunican con el host. Esto orquesta el sistema sin intervenir en el tráfico fino de spikes.
- **Software y mapeo.**  
Son toolchains para compilar/redesplegar SNNs a núcleos (particionado, asignación de neuronas/sinapsis, ruteo). Traduce modelos (por ejemplo, PyNN/Loihi API) a configuraciones de hardware cumpliendo límites de fan-in/fan-out, memoria y latencias.

Es decir que: Un sensor/event-encoder convierte señales en spikes. Los spikes entran a un núcleo; la crossbar/tabla sináptica suma entradas, actualiza neuronas y emite nuevos spikes. La NoC/AER reenvía esos spikes (multicast) a otros núcleos/chips. Si hay plasticidad, el motor local ajusta pesos en tiempo real. Los cores de control orquestan, miden y comunican con el host.

## 2.2. Ventajas y Desventajas

Ventajas	Desventajas
Alta eficiencia energética: consumen mucho menos que los computadores tradicionales para tareas de IA.	Tecnología inmadura: aún en fase de investigación y desarrollo, con aplicaciones limitadas.
Procesamiento paralelo masivo: inspirado en el cerebro humano, permite gran capacidad de cómputo en tareas cognitivas.	Alto costo de desarrollo: fabricación y diseño de chips neuromórficos es costosa.
Optimización en IA: muy eficientes en reconocimiento de patrones, visión artificial, procesamiento sensorial y robótica.	Dificultad de integración: falta de compatibilidad con arquitecturas clásicas de Von Neumann.
Escalabilidad biológica: permite crear redes que imitan al cerebro humano, facilitando avances en IA general.	Limitaciones en software: escasez de lenguajes y herramientas de programación adaptadas.
Robustez en condiciones adversas: algunos diseños toleran fallos y ruido en el procesamiento.	Aplicaciones poco claras a gran escala: su utilidad práctica frente a supercomputadores clásicos aún está en evaluación.

## 2.3. Tipos de computación neuromórfica

### ■ Computación Neuromórfica Digital:

Es hardware construido en tecnología CMOS “clásica” (digital) pero organizado como redes de neuronas que disparan picos (spikes). En lugar de hacer grandes multiplicaciones de matrices (como una GPU), estas máquinas procesan eventos: cada pico es un mensaje diminuto que viaja por una red-on-chip hacia sus sinapsis destino. Así se ahorra energía cuando no hay actividad y se imita la dinámica temporal de los sistemas biológicos. Una revisión técnica que enmarca el área (y compara enfoques analógico vs. digital) es la de Indiveri y Liu (Proc. IEEE, 2015). Se compone por:

- Neuronas y sinapsis digitales
- Memoria local cercana al cómputo
- Comunicación NoC / AER

### ■ Computación neuromórfica analógica:

En lugar de representar los estados neuronales como números discretos en hardware digital (como en TrueNorth o Loihi), aquí se usan señales analógicas (voltajes, corrientes) y elementos físicos que imitan directamente la dinámica de membranas y sinapsis biológicas. Puede ser analógica pura: todo el cálculo (suma de corrientes, integración de voltajes) ocurre con leyes físicas continuas (Ohm, Kirchhoff, capacitancias). o puede ser mixta: se usan bloques analógicos para la dinámica neuronal/sináptica, pero control digital para configuración, comunicación o calibración. Para su arquitectura se pueden comprender cuatro bloques de funcionamiento:

- Neuronas analógicas.  
Se implementan con transistores y condensadores que replican el modelo LIF (Leaky Integrate-and-Fire). Un capacitor acumula carga (potencial de membrana); cuando supera un umbral, un comparador genera un “spike”.
- Sinapsis analógica.  
Donde Resistores, transistores subumbral o memristores representan pesos sinápticos. La corriente que fluye depende del peso, lo que implementa la multiplicación peso  $\times$  spike de forma natural.
- Plasticidad local.  
STDP (Spike-Timing Dependent Plasticity) puede implementarse directamente en hardware: el solapamiento temporal de señales cambia la conductancia de un memristor o transistor.
- Comunicación.  
Muchos chips usan AER (Address-Event Representation), como en digital, pero el núcleo neuronal es analógico. Otras implementaciones son completamente analógicas, con redes de voltajes acoplados.

Conviene cuando se requiere eficiencia extrema (nanowatts por sinapsis), En prototipos de sistemas autónomos de bajo consumo (robots insectoides, sensores inteligentes) y sirve para estudiar modelos biológicos realistas con plasticidad emergente.

### ■ Computación neuromórfica Híbrida:

Este enfoque combina lo mejor de los dos mundos: Bloques digitales (para control, programación, escalabilidad y comunicación robusta), Bloques analógicos (para emular la dinámica neuronal y sináptica con alta eficiencia energética) y Algoritmos bioinspirados (modelos simplificados del cerebro como el aprendizaje local, redes oscilatorias, plasticidad, etc.). En este tipo, un chip híbrido suele dividirse en tres niveles:

- Capa analógica (neuronal/sináptica).  
Implementa la dinámica del potencial de membrana y las sinapsis (por ejemplo, con condensadores, transistores en subumbral o memristores). Aquí ocurre el cómputo físico (integración, disparo, plasticidad local).



- Capa digital (control y comunicación).  
Controla la configuración de parámetros (umbrales, pesos, tiempos de aprendizaje). Maneja la comunicación entre chips usando protocolos como AER (Address-Event Representation). Permite reprogramar redes sin necesidad de rediseñar el hardware.
- Software bioinspirado (modelos de red).  
Algoritmos que se acercan al comportamiento del cerebro, como aprendizaje Hebbiano, STDP, homeostasis. Traducción de redes de IA convencionales a redes spiking para aprovechar el hardware híbrido.
- Computación neuromórfica con Dispositivos emergentes:  
Este enfoque usa nuevos materiales y tecnologías de hardware diferentes al silicio tradicional, inspirados directamente en la forma en que funcionan las neuronas y sinapsis biológicas. En lugar de implementar neuronas y sinapsis solo con transistores CMOS clásicos, se aprovechan dispositivos como:
  - Memristores: Componentes que recuerdan la corriente pasada (sinapsis con memoria).
  - RRAM (Resistive RAM): Resistencias que cambian su valor según el voltaje aplicado.
  - PCM (Phase-Change Memory): Materiales que cambian entre estados amorfo y cristalino (como una sinapsis plástica).
  - Dispositivos spintrónicos: Usan el espín del electrón, no solo su carga, para procesar.
  - Nanofotónica: Usa la luz para transmitir información como si fueran spikes neuronales.
- Computación neuromórfica Fotónica:  
La computación neuromórfica fotónica utiliza luz (fotones) en lugar de electrones para procesar y transmitir información en sistemas que imitan al cerebro. En vez de transistores o memristores, se emplean láseres, moduladores ópticos, guías de onda y detectores de luz para construir "neuronas" y "sinapsis" artificiales. Funciona con una señal de entrada (un pulso de luz entra en un modulador óptico), luego ocurre un procesamiento neuromórfico, aquí la intensidad, fase o polarización de la luz se modifica (imitando cómo cambia la fuerza de una sinapsis), sigue una suma de señales en donde múltiples señales ópticas se combinan en un interferómetro, tal como lo hacen las neuronas al integrar impulsos. Por último ocurre una activación, es decir, un láser o detector convierte el resultado en una señal de salida óptica, lista para ser enviada a la siguiente "neurona".

### 3. Ordenador biológico

En sentido estricto, un ordenador biológico es un sistema de cómputo cuyo "hardware" incluye material biológico (p. ej., ADN/ARN, proteínas, células vivas u organoides neuronales) y que codifica, procesa y almacena información mediante procesos bioquímicos o biofísicos (no solo con transistores de silicio). El input suelen ser moléculas, luz o impulsos eléctricos; el output puede ser expresión génica, fluorescencia, secreción de moléculas o actividad eléctrica registrada en electrodos. Su atractivo es la paralelización masiva, la eficiencia energética y la capacidad de adaptación propia de los sistemas vivos.

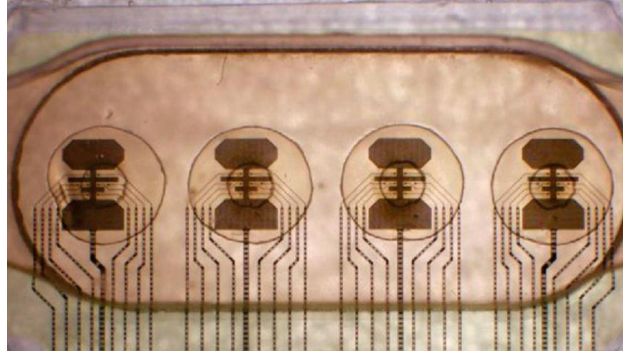


Figura 3: Ordenador biológico.

### 3.1. Arquitectura de un Ordenador biológico

- Hardware molecular.

- Portadores de información: secuencias de ADN/ARN (bits  $\rightarrow$  nucleótidos), niveles de proteínas (señales continuas) y estados químicos (fosforilaciones, metilaciones).
- Energía: ATP/GTP, gradientes iónicos y metabolismo celular (equivalente a la “fuente de poder”).
- Interconexión: difusión de moléculas dentro de la célula y entre células (por quorum sensing); en DNA-computing, hibridación por desplazamiento de cadenas (“strand displacement”). Ejemplos clásicos de circuitos de ADN puramente químicos muestran puertas lógicas y circuitos más grandes contruidos solo con oligos de ADN.

- Capa de dispositivos (primitivas lógicas y sensores).

Dispositivos básicos que actúan como puertas lógicas o sensores:

- Promotores/represores: activan o silencian genes (NOT, NOR, AND con combinaciones).
- CRISPRi (dCas9): una nucleasa “muerta” que se guía con sgRNA para reprimir un promotor; al modular varios sgRNA se implementan NOR apilables. Se han construido NOR en levadura y bacterias con conexiones arbitrarias.
- Ribocomputing (ARN): toehold switches y ARN diseñados que se reconocen entre sí y abren/cierran la traducción; con esto se han hecho compuertas AND/OR y expresiones lógicas con hasta 12 entradas dentro de *E. coli*.
- ADN de desplazamiento de cadenas (sistemas libres de células): puertas digitales/neuronales en tubo de ensayo; por ejemplo, un circuito que calcula raíz cuadrada de 4 bits (130 hebras).

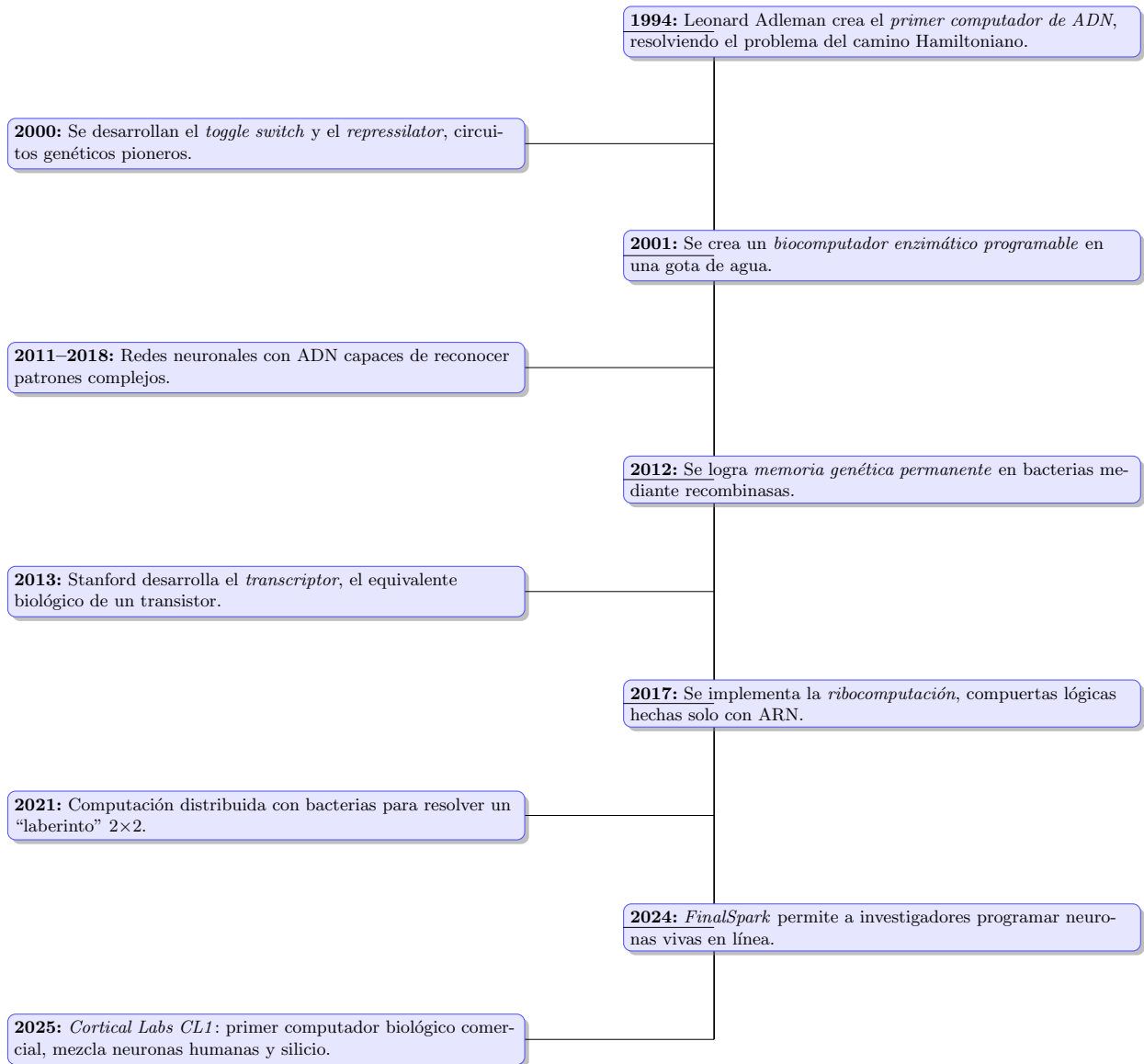
- Capa de circuitos (combinacionales y secuenciales).

- Combinacionales: se encadenan NOR/NOT/AND para decidir salidas (expresión de un gen reportero, producción de un metabolito, etc.). CRISPRi y ribocomputing permiten fan-in alto (varias entradas por compuerta).
- Secuenciales / memoria:
  - Biestables (toggle switch): dos genes que se reprimen mutuamente; almacenan 1 bit estable “ON/OFF” y se conmutan con un pulso químico. Es el clásico interruptor genético en *E. coli*.
  - Registro/bitácora con CRISPR: se han escrito imágenes y un “minivideo” dentro del genoma bacteriano usando adquisición de espaciadores (Cas1-Cas2), demostrando memoria cronológica en ADN.

En ADN sintético (sin células), las cascadas de desplazamiento de cadenas también implementan memoria y redes tipo neuronal (umbral) a partir de reacciones químicas.

- Capa de sistema (chasis y comunicación).
  - Chasis: bacterias (por ejemplo: *E. coli*), levadura, células de mamífero, o sistemas libres de células (transcripción-traducción in vitro).
  - Redes multicelulares: comunicación por señales difusibles; se reparten funciones entre cepas (dividir la lógica para escalar y reducir carga metabólica).
  - Espacio/tiempo: el “reloj” es la cinética de transcripción/traducción (minutos-horas); la difusión crea cables químicos de baja velocidad.
- 5) Capa de programación.
  - Lenguaje de bajo nivel: secuencias de promotores, RBS, terminadores, guías CRISPR, toeholds; constantes de unión y tasas de expresión.
  - Compiladores CAD: herramientas como Cello toman una especificación booleana y devuelven un diseño de circuito genético (partes y secuencias) listo para clonar y probar.
  - Verificación: se simulan modelos (ODE/estocásticos) y se caracterizan transferencias (curvas entrada-salida) para ajustar umbrales.
- Entrada/Salida.
  - Entradas: pequeñas moléculas (arabinosa, IPTG), luz (optogenética), temperatura, pH, metabolitos, ARN específicos, incluso señales ambientales.
  - Salidas: fluorescencia (GFP), color, crecimiento/muerte, secreción de moléculas, cambios de ADN (memoria), o un fenotipo útil (por ejemplo, producir un fármaco).

### 3.2. Principales hitos de los ordenadores biológicos



## 4. Computación Heterogénea

La computación heterogénea es un paradigma de cómputo en el que se combinan diferentes tipos de procesadores o unidades de cómputo dentro de un mismo sistema, con el fin de aprovechar las fortalezas específicas de cada uno y obtener mayor rendimiento, eficiencia energética y flexibilidad.

Un ordenador homogéneo tradicional usa un único tipo de procesador (por ejemplo, solo una CPU). En cambio, en un sistema heterogéneo se utilizan varios tipos de procesadores, como:

- CPU (Unidad Central de Procesamiento): buena en tareas secuenciales y de propósito general.
- GPU (Unidad de Procesamiento Gráfico): excelente en cómputo paralelo masivo.
- FPGA (Arreglos de compuertas programables en campo): hardware reconfigurable para tareas específicas.
- TPU (Tensor Processing Unit): aceleradores diseñados para inteligencia artificial.

- DSP (Procesadores digitales de señales): optimizados para procesamiento de audio, video y señales en tiempo real.

La idea es que cada tarea se ejecute en la unidad que mejor se adapta a su naturaleza, logrando mayor rendimiento y eficiencia.

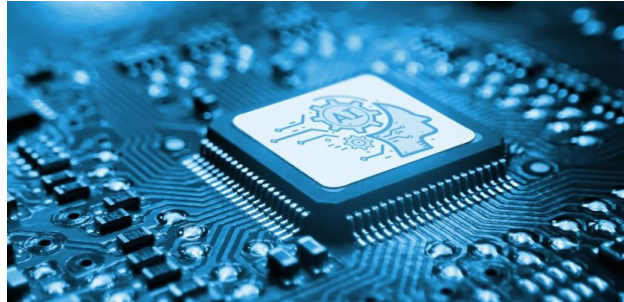


Figura 4: Computación heterogénea.

#### 4.1. Historia de la computación heterogénea

La evolución de la computación heterogénea se puede resumir en los siguientes hitos:

- **1960-1980: Primeros pasos** Surgen los supercomputadores (ej. Cray-1 en 1976) y coprocesadores matemáticos (Intel 8087 en 1980). Se introduce la idea de usar distintos procesadores para distintas tareas.
- **1990: GPUs y aceleradores** Aparecen las primeras *GPUs* enfocadas en gráficos, que poco después se empiezan a usar para cálculos generales (*GPGPU*).
- **2000s: Formalización de la computación heterogénea** NVIDIA lanza **CUDA** en 2006 y Khronos Group presenta **OpenCL** en 2009. Surgen chips heterogéneos como el *Cell Broadband Engine* (IBM, 2005) y las **APUs** de AMD (2011).
- **2010s: Expansión con IA y Big Data** Las GPUs se consolidan en **Inteligencia Artificial** y **Big Data**. Google introduce sus **TPUs** (2016) y Microsoft usa **FPGAs** en Azure. Supercomputadores como *Summit* (2018) combinan CPU + GPU.
- **2020 en adelante: Heterogeneidad masiva** Surgen arquitecturas híbridas de gran escala, como el supercomputador *Frontier* (2022, Oak Ridge), basado en CPU + GPU. Los SoCs móviles integran CPU + GPU + NPU. Se investiga la integración con cómputo cuántico y neuromórfico.

## 4.2. Ventajas y Desventajas

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>■ Alto rendimiento al combinar diferentes tipos de procesadores.</li> <li>■ Mejor eficiencia energética, ya que cada unidad ejecuta la tarea para la que está optimizada.</li> <li>■ Flexibilidad para diferentes aplicaciones: IA, simulaciones científicas, gráficos, etc.</li> <li>■ Escalabilidad: integración en supercomputadores y dispositivos móviles.</li> </ul>	<ul style="list-style-type: none"> <li>■ Mayor complejidad de programación y optimización.</li> <li>■ Dependencia de frameworks específicos (CUDA, OpenCL, etc.).</li> <li>■ Portabilidad limitada entre arquitecturas heterogéneas.</li> <li>■ Costos elevados en hardware especializado.</li> </ul>

## 5. Arquitectura de computación de borde

La computación de borde (Edge Computing) es un modelo de arquitectura computacional en el que el procesamiento de datos no se realiza principalmente en la nube ni en un centro de datos centralizado, sino cerca de la fuente donde se generan los datos (es decir, en el “borde” de la red).

En vez de enviar toda la información a servidores lejanos para procesarla, los dispositivos o nodos de borde (como gateways, routers inteligentes, sensores avanzados, microservidores o incluso el propio dispositivo del usuario) realizan parte del procesamiento localmente.

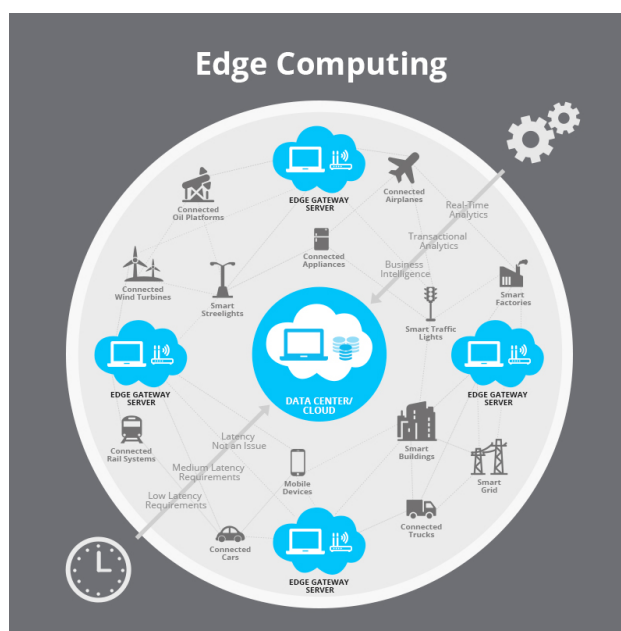


Figura 5: Cómo es la computación de borde.

### 5.1. Historia de la Computación de borde

La **computación de borde** (*Edge Computing*) surge como una respuesta a las limitaciones del modelo tradicional de la nube, el cual depende de centros de datos centralizados. Su evolución puede entenderse en

varias etapas:

- **Década de 1990:** Aparecen las primeras redes de distribución de contenido (CDN, por sus siglas en inglés), diseñadas para acercar copias de archivos estáticos (como imágenes o páginas web) a los usuarios, reduciendo la latencia y mejorando la experiencia de navegación.
- **Década de 2000:** Con la masificación de Internet y la aparición de servicios en la nube, se hace evidente que el envío de grandes volúmenes de datos hacia centros de datos distantes genera cuellos de botella y altos costos de ancho de banda. Se empieza a considerar la necesidad de un procesamiento más descentralizado.
- **Década de 2010:** El auge del **Internet de las Cosas (IoT)** y la industria 4.0 acelera el interés por llevar el procesamiento cerca de la fuente de datos. Sensores, cámaras y dispositivos inteligentes requieren respuestas en tiempo real, lo que impulsa la arquitectura de computación de borde.
- **Actualidad:** La computación de borde se integra con tecnologías como **5G, inteligencia artificial distribuida y cloud híbrida**, lo que permite aplicaciones críticas en áreas como vehículos autónomos, salud digital, ciudades inteligentes y sistemas de seguridad. Hoy en día, es un componente esencial en la infraestructura tecnológica moderna.

En síntesis, la computación de borde representa una evolución natural desde la distribución de contenido en los años noventa hasta convertirse en un pilar de la transformación digital en la actualidad, complementando y optimizando los servicios de la nube.

## 5.2. Ventajas y Desventajas

Ventajas	Desventajas
<ul style="list-style-type: none"><li>■ <b>Baja latencia:</b> al procesar los datos cerca de la fuente, las respuestas son casi inmediatas.</li><li>■ <b>Optimización del ancho de banda:</b> solo se envían a la nube los datos relevantes.</li><li>■ <b>Mayor resiliencia:</b> los sistemas pueden seguir operando incluso con conexión limitada a la nube.</li><li>■ <b>Escalabilidad distribuida:</b> se pueden desplegar nodos en diferentes ubicaciones.</li><li>■ <b>Seguridad mejorada:</b> al mantener ciertos datos sensibles localmente, se reduce el riesgo de exposición.</li></ul>	<ul style="list-style-type: none"><li>■ <b>Complejidad en la gestión:</b> administrar múltiples nodos distribuidos requiere mayor esfuerzo y coordinación.</li><li>■ <b>Costos iniciales altos:</b> se necesita invertir en infraestructura local (hardware, software, mantenimiento).</li><li>■ <b>Seguridad heterogénea:</b> cada nodo puede tener vulnerabilidades distintas que aumentan la superficie de ataque.</li><li>■ <b>Escalabilidad limitada localmente:</b> algunos dispositivos de borde tienen poca capacidad de cómputo o almacenamiento.</li><li>■ <b>Estandarización en desarrollo:</b> aún falta madurez en protocolos y marcos de referencia.</li></ul>

## 6. REFERENCIAS.

1. IBM Quantum. (2023). *What is quantum computing?* Disponible en: <https://www.ibm.com/quantum>
2. Microsoft Quantum. (2023). *Introduction to quantum computing.* Disponible en: <https://learn.microsoft.com/en-us/azure/quantum/overview-quantum-computing>

3. Quantum Country. (2022). *An introduction to quantum computing*. Disponible en: <https://quantum.country/qcvc>
4. Intel Labs. (2023). *Intel Loihi neuromorphic research chip*. Disponible en: <https://www.intel.com/content/www/us/en/research/neuromorphic-computing.html>
5. IBM Research. (2023). *TrueNorth neuromorphic chip*. Disponible en: <https://research.ibm.com/topics/neuromorphic-computing>
6. Nature News. (2021). *Neuromorphic computing: why it matters*. Disponible en: <https://www.nature.com/articles/d41586-021-02212-2>
7. MIT Technology Review. (2021). *The rise of biological computers*. Disponible en: <https://www.technologyreview.com/2021/02/12/1017771/biological-computers-dna-protein/>
8. Wyss Institute (Harvard). (2020). *Synthetic biology and biological computing*. Disponible en: <https://wyss.harvard.edu/technology/synthetic-biology/>
9. National Human Genome Research Institute. (2022). *DNA computing*. Disponible en: <https://www.genome.gov/about-genomics/fact-sheets/DNA-Computing-Fact-Sheet>
10. IBM. (2023). *Edge Computing Explained*. Disponible en: <https://www.ibm.com/topics/edge-computing>
11. Cisco. (2021). *What is edge computing?* Disponible en: <https://www.cisco.com/c/en/us/solutions/edge-computing/overview.html>
12. Microsoft Azure. (2023). *Introduction to edge computing*. Disponible en: <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/iot-edge/iot-edge>
13. NVIDIA Developer. (2023). *What is heterogeneous computing?* Disponible en: <https://developer.nvidia.com/heterogeneous-computing>
14. AMD Developer Central. (2023). *Heterogeneous System Architecture (HSA)*. Disponible en: <https://developer.amd.com/resources/heterogeneous-computing/>
15. Top500. (2022). *Heterogeneous computing in modern supercomputers*. Disponible en: <https://www.top500.org/news/heterogeneous-computing/>