



Universitat d'Alacant  
Universidad de Alicante

## Retos futuros de la Computación de Alto Rendimiento (CAR)

---

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



## Indice

---

### Contenido

Tendencias futuras en Computación de Alto Rendimiento (CAR) .....	4
Resumen sobre la lectura de hoy: .....	6
1. Procesadores híbridos (CPU+GPU+IA) .....	6
2. Supercomputadores heterogéneos .....	6
3. Redes de ultra baja latencia .....	6
4. Co-diseño HPC + Inteligencia Artificial .....	6
5. Software inteligente y autoadaptativo .....	7
1.1 ¿Qué son los procesadores híbridos? .....	8
1.1.1 Definición de NPU y TPU .....	8
1.2. ¿Por qué han surgido estas arquitecturas? .....	9
1.3. Ejemplos actuales destacados .....	9
1.4. Resumen comparativo .....	11
1.4. Procesadores híbridos .....	13
1.5. Distribución de tareas entre unidades .....	13
1.6 Consideraciones sobre privacidad y autonomía .....	15
2. Modelos heterogéneos de computación .....	16
2.1 Evolución histórica .....	16
2.2 Computación cuántica .....	17
2.3. Procesadores fotónicos .....	20
Introducción .....	20
Parte electrónica .....	21
2.4. Chips neuromórficos .....	26
2.4.1. Definición general .....	26
2.4.2. Fundamentos técnicos .....	27
2.4.3. Fabricantes principales y ejemplos relevantes .....	28
2.4.4. Aplicaciones típicas .....	29
2.4.5. Ventajas (resumen) .....	29
2.4.6. Limitaciones .....	29
2.4.7. Convivencia con arquitecturas clásicas .....	30
3. Redes de Ultra Baja Latencia .....	30



# Universitat d'Alacant Universidad de Alicante

3.1. Introducción.....	30
3.2. Tecnologías Emergentes .....	30
3.2.1. CXL (Compute Express Link) .....	30
3.2.2. Infiniband HDR (High Data Rate).....	31
3.2.3. Quantum Networking .....	32
4. Software inteligente y autoadaptativo.....	33
Bibliografía y enlaces recomendados.....	40

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



## Parte 0: Presentación de la sesión

---

### Tendencias futuras en Computación de Alto Rendimiento (CAR)

*“El futuro de la computación no solo se mide en flops o núcleos, sino en su capacidad para adaptarse, reducir su impacto ecológico y democratizar el acceso al poder de cálculo.”*

En esta sesión exploraremos cómo están evolucionando los sistemas de computación de alto rendimiento y qué retos plantea este desarrollo en términos de sostenibilidad, arquitectura, automatización e inteligencia artificial.

### Ámbitos de reflexión de hoy

- Procesadores híbridos y supercomputadores heterogéneos.
- Energía, latencia y eficiencia en entornos distribuidos.
- Uso de IA para el diseño, la gestión y la resiliencia de sistemas HPC.
- Rol del software autoadaptativo en los sistemas exa- y zetta-escala.

### Objetivo de la sesión

Comprender no solo las tecnologías que marcarán el futuro de la CAR, sino también las implicaciones de su adopción:

¿Quién decide cómo se diseña un sistema inteligente? ¿Quién lo controla? ¿Qué consecuencias tiene para la sociedad y el planeta?

### ¿Cómo se va a desarrollar la sesión?

Durante esta sesión vamos a trabajar de forma activa los contenidos que habéis leído previamente sobre los retos futuros en la Computación de Alto Rendimiento (CAR). No se trata de una clase expositiva tradicional, sino de una actividad colaborativa donde el conocimiento se construye a partir de vuestras respuestas, preguntas y reflexiones.

### La dinámica será la siguiente:

1. **Lectura inicial (10-15 min):**  
Comenzaremos con unos minutos de lectura individual para repasar el resumen de los cinco bloques principales. Podéis subrayar ideas clave o marcar aquellas partes que os resulten más confusas o interesantes.
2. **Preguntas del profesor por bloques temáticos (70 min aprox.):**  
A continuación, iremos analizando los contenidos por partes. Yo iré lanzando preguntas sobre lo que habéis leído, y a partir de vuestras respuestas iremos

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



Universitat d'Alacant  
Universidad de Alicante

aclarando conceptos, explicando ejemplos y profundizando en las ideas más importantes.

3. **Reflexión final y debate abierto (30-35 min):**

Terminaremos con un pequeño debate grupal guiado, a partir de algunas preguntas críticas del documento. El objetivo es que conectéis los contenidos con situaciones reales, dilemas éticos o futuros posibles.



# Parte 1: Tendencias Futuras en Computación de Alto Rendimiento (CAR)

---

## Resumen sobre la lectura de hoy:

### 1. Procesadores híbridos (CPU+GPU+IA)

- Integración de componentes especializados en el mismo chip.
- Mejoran eficiencia energética y aceleran aplicaciones de IA, simulaciones, Big Data.
- Ejemplos: NVIDIA Grace Hopper, AMD MI300.

### 2. Supercomputadores heterogéneos

- Combinan arquitecturas clásicas (CPU) con tecnologías emergentes:
  - Computación cuántica (IBM, Google).
  - Procesadores fotónicos (Lightmatter).
  - Chips neuromórficos (Intel Loihi).
- El objetivo es atacar distintos tipos de problemas de forma más eficiente.

### 3. Redes de ultra baja latencia

- **Tecnologías emergentes:** CXL (Compute Express Link), Infiniband HDR, Quantum Networking.
- **Importancia:** minimizar cuellos de botella entre procesadores y almacenamiento.

### 4. Co-diseño HPC + Inteligencia Artificial

- Aplicación de IA para:
  - Optimizar diseños de sistemas HPC.
  - Predecir fallos y ajustar dinámicamente configuraciones.
- Ejemplo: DeepMind + Google Data Centers.



Universitat d'Alacant  
Universidad de Alicante

## 5. Software inteligente y autoadaptativo

- Sistemas de gestión que:
  - Se autorregulan según cargas y errores.
  - Reducen intervención humana.
- Importancia para la resiliencia de los futuros exa- y zetta-computadores.



## 1.1 ¿Qué son los procesadores híbridos?

Los procesadores híbridos son arquitecturas avanzadas que integran, en un único chip o sistema compacto, múltiples unidades de procesamiento especializadas para diferentes tipos de tareas:

- **CPU (Unidad Central de Proceso):** Adecuada para tareas secuenciales, decisiones lógicas y control del sistema operativo y las aplicaciones.
- **GPU (Unidad de Procesamiento Gráfico):** Diseñada para ejecutar miles de operaciones en paralelo, ideal para gráficos, simulaciones físicas y procesamiento de datos masivos.
- **Aceleradores de IA (como NPUs o TPUs):** Orientados a tareas específicas de inteligencia artificial, especialmente la inferencia de modelos ya entrenados y el aprendizaje profundo.

Esta combinación permite abordar problemas computacionales modernos con un enfoque más eficiente, tanto en rendimiento como en consumo energético.

### 1.1.1 Definición de NPU y TPU

Las **NPU** (Neural Processing Unit) y las **TPU** (Tensor Processing Unit) son tipos específicos de procesadores diseñados para acelerar tareas relacionadas con inteligencia artificial, especialmente para cálculos asociados con redes neuronales profundas y aprendizaje automático.

#### NPU (Neural Processing Unit)

La **NPU** es un procesador optimizado específicamente para realizar operaciones relacionadas con redes neuronales artificiales, como inferencia o procesamiento en tiempo real.

Es ideal para aplicaciones de inferencia en dispositivos portátiles o integrados, donde se busca eficiencia energética

#### Características principales:

- **Optimización específica:** Realiza cálculos de forma extremadamente eficiente para operaciones frecuentes en redes neuronales, como multiplicación matricial y convoluciones.
- **Bajo consumo de energía:** Usado especialmente en dispositivos móviles (smartphones, tabletas, dispositivos IoT) para mantener una alta eficiencia energética.
- **Uso principal:** Inferencia de inteligencia artificial (IA) en aplicaciones móviles y embebidas.





## Ejemplos prácticos:

- Procesadores **Apple Neural Engine** en chips A-series y M-series.
- **Huawei NPU** integrada en los chips Kirin.
- **Samsung Exynos** con unidades dedicadas de IA.

## TPU (Tensor Processing Unit)

Las **TPU** son aceleradores de hardware diseñados específicamente por Google para acelerar el entrenamiento e inferencia en modelos de aprendizaje automático basados en tensores (estructuras de datos multidimensionales usadas por bibliotecas como TensorFlow).

## Características principales:

**Optimizado para TensorFlow:** Aunque pueden trabajar con otras plataformas, fueron inicialmente diseñadas para optimizar el rendimiento con esta biblioteca.

- Altísima velocidad en cálculos de matrices y tensores.
- Ideal para entrenamiento profundo y procesamiento en centros de datos.

## 1.2. ¿Por qué han surgido estas arquitecturas?

El surgimiento de los procesadores híbridos responde a varios factores estructurales y tecnológicos:

**Estancamiento en la mejora del rendimiento por núcleo en CPU** tradicionales. La Ley de Moore ha perdido su ritmo, haciendo necesario explotar nuevas formas de paralelismo.

**Crecimiento exponencial de los datos y de la complejidad de las aplicaciones.** La inteligencia artificial, el modelado físico, el análisis de datos en tiempo real y los videojuegos hiperrealistas exigen nuevas capacidades.

**Cambio de paradigma: de arquitecturas generalistas a sistemas heterogéneos.** En lugar de confiar en una única arquitectura universal, se opta por integrar unidades especializadas que trabajen en colaboración.

## 1.3. Ejemplos actuales destacados

Podemos distinguir claramente entre las utilizadas a nivel SuperCluster y Centros de datos y las utilizadas a nivel Doméstico

## Nivel Supercluster y Centros de Datos

### 1. Google Ironwood (TPU)

- Uso principal: **Inferencia a gran escala** (LLMs, MoE, modelos generativos).
- Características destacadas:
  - 7ª generación de TPU (2025).
  - 192 GB memoria HBM por chip.

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



- 7.2 TB/s ancho de banda por chip.
- Escalable hasta 42.5 exaFLOPS por pod (9216 chips).
- Optimizada para eficiencia energética y velocidad en inferencia masiva.
- Lectura recomendada: <https://blog.google/products/google-cloud/ironwood-tpu-age-of-inference>

## 2. NVIDIA H100 Tensor Core (GPU)

- Uso principal: Entrenamiento y cargas mixtas (inferencia/entrenamiento).
- Características destacadas:
  - Arquitectura Hopper.
  - Alto rendimiento en entrenamiento de modelos complejos.
  - Escalable en grandes clusters y centros de datos.
  - Soporte para formatos de precisión mixtos y cálculos avanzados (Transformer Engines).

## 3. NVIDIA A100 Tensor Core (GPU)

- Uso principal: Entrenamiento, inferencia a gran escala, HPC.
- Características destacadas:
  - Arquitectura Ampere.
  - Amplia adopción actual en centros de datos.
  - Equilibrio entre rendimiento energético y computacional.

## 4. NVIDIA B100/B200 Blackwell (GPU) (emergente)

- Uso principal: IA generativa, entrenamiento masivo.
- Características destacadas:
  - Nueva generación Blackwell (2025).
  - Optimización energética avanzada.
  - Innovación en cálculos específicos para modelos generativos.

## Nivel Doméstico y Estaciones de Trabajo

### 1. NVIDIA GeForce RTX 4090

- Uso principal: Entrenamiento e inferencia local, diseño gráfico, juegos avanzados.
- Características destacadas:
  - Arquitectura Ada Lovelace.
  - Alta capacidad VRAM (24 GB).
- Ideal para desarrolladores y entusiastas que ejecutan modelos de IA localmente.



### 2. NVIDIA RTX A6000

- Uso principal: Profesionales de IA, renderizado, desarrollo de modelos.
- Características destacadas:
  - 48 GB GDDR6 VRAM.
  - Excelente soporte profesional, certificaciones para aplicaciones profesionales.

### 3. AMD Radeon RX 7900 XTX

- Uso principal: Inferencia y entrenamiento ligero, juegos y creación de contenido.
- Características destacadas:
  - Arquitectura RDNA 3.
  - Buen rendimiento precio/calidad, alternativa a NVIDIA.

## 1.4. Resumen comparativo

- A **nivel supercluster** y grandes centros de datos, las **TPUs (Ironwood)** compiten directamente con GPUs avanzadas como **NVIDIA H100/A100** por rendimiento en tareas especializadas, siendo Ironwood particularmente fuerte para inferencia masiva.
- A **nivel doméstico**, **NVIDIA mantiene liderazgo absoluto**, especialmente con RTX 4090/A6000. AMD es una alternativa económica y eficiente para algunas aplicaciones menos intensivas.

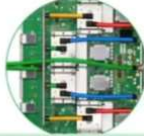


Tabla comparativa:

Modelo	Arquitectura / Tipo	Memoria	Uso Principal	Eficiencia Energética	Rendimiento Estimado	Precio Aproximado (USD)
Google TPU Ironwood	TPU (7ª generación)	192 GB HBM	Inferencia a gran escala	Muy alta	4,614 TFLOPS	N/D (servicio en la nube)
NVIDIA H100	Hopper (GPU)	80 GB HBM2e	Entrenamiento e inferencia	Alta	4,000 TFLOPS (FP8), ~1,000 TFLOPS (FP16)	\$25,000 – \$40,000
NVIDIA A100	Ampere (GPU)	40/80 GB HBM2e	Entrenamiento e inferencia	Media	624 TFLOPS (FP16 Tensor)	\$7,999 – \$17,549
NVIDIA RTX A6000	Ampere (GPU)	48 GB GDDR6	Estaciones de trabajo	Media	75 TFLOPS (FP16 Tensor)	\$4,586 – \$8,300
NVIDIA RTX 4090	Ada Lovelace (GPU)	24 GB GDDR6X	Uso doméstico y profesional	Media	330 TFLOPS (FP16 Tensor)	\$2,500 – \$3,500



AMD	RDNA 3 (GPU)	24 GB	Uso doméstico	Alta	123 TFLOPS	\$890 – \$1,089
Radeon RX 7900 XTX		GDDR6			(FP16)	

Comparativa evolutiva entre versiones de TPU de Google:

	 TPU v4	 TPU v5p	 Ironwood
	2022	2023	2025
Pod Size (chips)	4096	8960	9216
HBM Bandwidth/ Capacity	32 GB @ 1.2 TBps HBM	95 GB @ 2.8 TBps HBM	192 GB @ 7.4 TBps HBM
Peak Flops per chip	275 TFLOPS	459 TFLOPS	4614 TFLOPS

**NOTA:** ¿Porqué esas diferencias de precio **entre modelos iguales**?

La gran variabilidad de precios surge de diferencias en configuraciones técnicas, versiones especializadas para distintos tipos de clientes, fluctuaciones en demanda, ubicación geográfica y servicios adicionales asociados.

Por ello, siempre es importante especificar claramente la versión exacta del producto cuando se compara o compra una GPU profesional, para asegurarte que el precio pagado coincide con el valor real del producto según tus necesidades específicas.

**Ejemplo práctico con NVIDIA H100:**

Versión	Precio aproximado	Explicación
<b>H100 PCIe 80GB (Nueva)</b>	~\$25,000	Versión estándar PCIe, menos rendimiento térmico y energético.
<b>H100 SXM5 80GB (Nueva)</b>	~\$35,000 \$40,000	- Mayor rendimiento, sistema especializado para centros de datos y supercomputación.
<b>H100 PCIe 80GB (Reacondicionada)</b>	~\$20,000 \$24,000	- Menos garantía y soporte técnico limitado.

### Enlaces recomendados

- Google ganando la carrera de la IA con “Tensor Processing Unit”:  
[https://www.youtube.com/live/Pwbncs\\_Q9hc?feature=shared](https://www.youtube.com/live/Pwbncs_Q9hc?feature=shared)



- **Introducción a Cloud TPU:** Una guía detallada sobre qué son las TPUs y cómo se utilizan en Google Cloud:  
<https://cloud.google.com/tpu/docs/intro-to-tpu?hl=es>
- **Unidades de procesamiento tensorial (TPU) - Google Cloud:** Información sobre las versiones disponibles de TPUs y sus aplicaciones en el entrenamiento e inferencia de modelos de IA.

[https://cloud.google.com/tpu?hl=es-419&utm\\_source=chatgpt.com](https://cloud.google.com/tpu?hl=es-419&utm_source=chatgpt.com)

## 1.4. Procesadores híbridos

Los procesadores híbridos no son simplemente “chips más grandes”. Su diseño implica avances en:

- **Tecnología de chiplets:** Permite ensamblar módulos especializados en una sola unidad lógica, con flexibilidad en la fabricación y la integración.
- **Sistemas de interconexión de alta velocidad:** Como NVLink o Infinity Fabric, que garantizan un intercambio de datos fluido entre las unidades.
- **Memoria jerárquica compartida:** HBM (High Bandwidth Memory) y accesos directos entre CPU-GPU reducen el cuello de botella habitual en arquitecturas separadas.

El diseño se enfoca no solo en velocidad, sino también en minimizar la energía por operación, clave en la sostenibilidad de los centros de datos.

## 1.5. Distribución de tareas entre unidades

En un procesador híbrido bien diseñado, cada componente tiene un rol distinto que no se solapa, sino que se complementa:

La **CPU** gestiona el sistema operativo, organiza los procesos, maneja excepciones y controla la ejecución de programas complejos con secuencias condicionales.

La **GPU** ejecuta bloques de datos paralelizables, como matrices o vectores, mediante miles de hilos que operan de forma simultánea.

La **IA embebida** ejecuta inferencias directamente, sin necesidad de pasar por la CPU o GPU. Además, puede aprender patrones de carga y predecir optimizaciones, actuando como gestor autónomo de recursos en tiempo real.

### Ejemplo práctico: reconocimiento facial en un smartphone

**Desbloqueo de un móvil mediante reconocimiento facial:** Este proceso involucra distintas tareas que se distribuyen de forma complementaria entre la CPU, GPU y la IA embebida.



## 1. La CPU (Unidad Central de Procesamiento):

- **Rol:** Control, gestión del sistema y lógica compleja
  - Activa el sistema de cámara frontal cuando detecta que el usuario levanta el teléfono.
  - Coordina los procesos necesarios para iniciar el reconocimiento facial.
  - Verifica que el sistema esté en un entorno seguro (pantalla activa, sin llamadas, batería suficiente).
  - Decide qué módulo debe encargarse de cada tarea: delega procesamiento de imagen a la GPU y la inferencia a la NPU.
  - Equivalente al gestor principal que reparte tareas y toma decisiones.

## 2. La GPU (Unidad Gráfica)

- **Rol:** Cálculo intensivo y paralelo
  - Procesa la imagen de la cámara frontal, transformando los píxeles en una representación vectorial (mapa de características del rostro).
  - Realiza operaciones matemáticas pesadas (matrices, vectores, convoluciones) en paralelo.
  - Convierte la imagen en datos estructurados listos para la IA.
  - Realiza un trabajo intensivo y rápido.

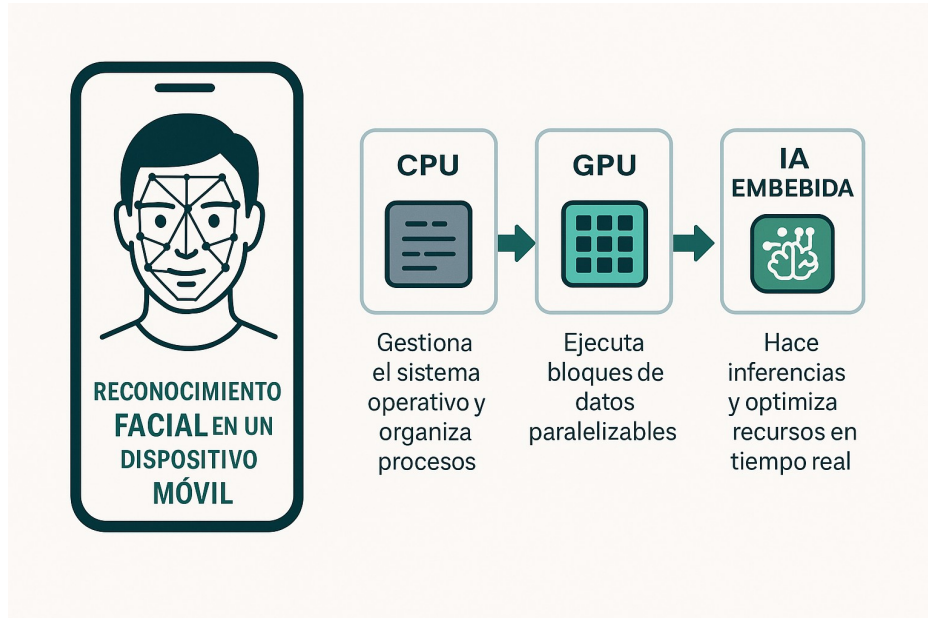
## 3. La unidad de IA embebida (por ejemplo, una NPU o TPU móvil)

- **Rol:** Toma de decisiones autónomas basadas en IA
  - Recibe los vectores procesados por la GPU y ejecuta el modelo de reconocimiento facial previamente entrenado.
  - Compara tu rostro con el perfil guardado en el dispositivo.
  - Decide en milisegundos si eres tú o no.
  - Además, aprende con el uso: si siempre desbloqueas el móvil en ciertos ángulos o condiciones de luz, ajusta dinámicamente la sensibilidad y consumo energético.
  - Es un asistente inteligente que no solo actúa, sino que aprende y mejora con la experiencia.





Los diferentes roles puedes verlos en el siguiente gráfico:



Cada componente hace lo que mejor sabe hacer, sin duplicar funciones:

Componente	Tareas asignadas	No hace...
<b>CPU</b>	Control general, decisiones, gestión del SO	No ejecuta operaciones pesadas en paralelo
<b>GPU</b>	Procesamiento de imagen y cálculos paralelos	No decide si desbloquear o no el móvil
<b>IA embebida</b>	Inferencia rápida, autónoma y optimización	No gestiona la cámara ni coordina el sistema

En un sistema híbrido bien diseñado, como un móvil moderno o un portátil con IA acelerada por hardware, **cada unidad realiza tareas especializadas**. Esto mejora la velocidad, reduce el consumo de energía y permite respuestas más inteligentes y rápidas.

## 1.6 Consideraciones sobre privacidad y autonomía

Uno de los aportes clave de la IA embebida en estos sistemas es la posibilidad de realizar tareas sensibles sin conexión a internet ni transferencia de datos:

### Ventajas:

- Menor exposición a vulnerabilidades.
- Reducción del consumo de ancho de banda.
- Ejecución más rápida y sin latencia de red.



## *Limitaciones:*

- El hecho de que el procesamiento sea local no garantiza que los datos derivados (como identificadores, hashes o resultados) no sean almacenados o enviados posteriormente.
- Depende del diseño y la política de cada fabricante.

## 2. Modelos heterogéneos de computación

Tras la consolidación de arquitecturas híbridas en dispositivos personales y edge computing (ver apartado 1), los centros de datos y supercomputadores también han migrado progresivamente hacia modelos heterogéneos a gran escala.

Este cambio no solo implica el uso coordinado de CPU, GPU y aceleradores de IA, sino también una nueva lógica de interconexión, escalabilidad y optimización energética en clústeres distribuidos.

### 2.1 Evolución histórica

Los sistemas heterogéneos de próxima generación no solo combinan CPU y GPU, sino que integran aceleradores basados en paradigmas no clásicos.

A lo largo de la historia de la computación, la evolución tecnológica ha seguido un patrón recurrente: primero surge una nueva arquitectura general (como la CPU), y posteriormente aparecen componentes especializados (como GPU, NPU o TPU) que la complementan y amplían sus capacidades.

En este punto del desarrollo, nos encontramos ante una situación similar. Tecnologías emergentes como **la computación cuántica, la fotónica o los chips neuromórficos** representan hoy lo que fueron las primeras CPU hace décadas: propuestas radicalmente nuevas, aún en fase experimental, pero con el potencial de transformar la computación de alto rendimiento.

Estas tecnologías no están diseñadas para sustituir a las CPU o GPU tradicionales, sino para **resolver de forma más eficiente tipos de problemas que los sistemas actuales abordan con dificultad**. Su integración futura permitirá construir sistemas heterogéneos aún más versátiles, capaces de combinar diferentes arquitecturas según la naturaleza del cálculo requerido.

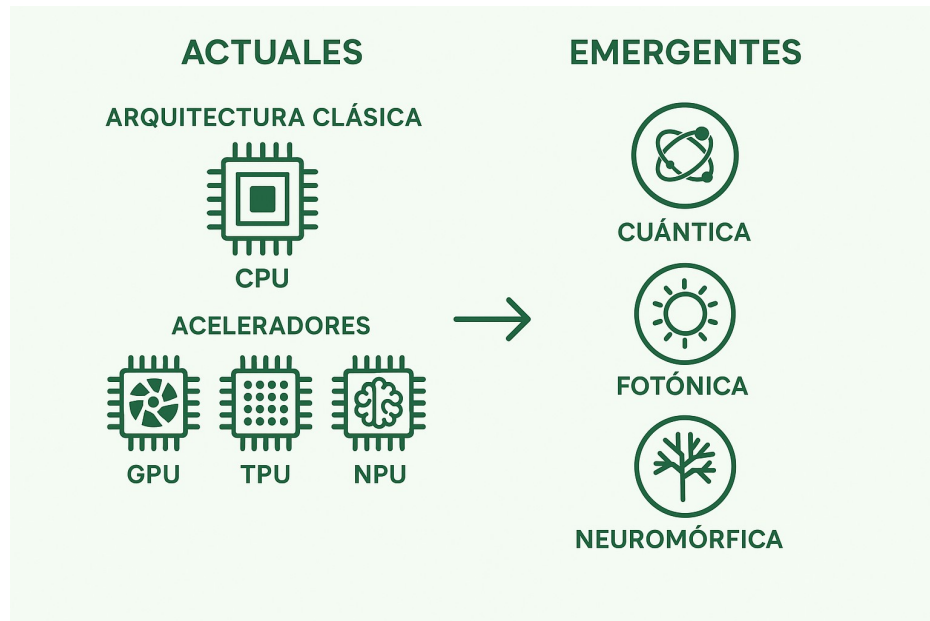
### *¿Qué significa esto?*

- Las tecnologías emergentes actuales (como los chips cuánticos o fotónicos) están en la misma **posición que estaban las CPU hace 50 años**.
- Aún **no tienen ecosistema maduro**, pero están diseñadas para resolver tipos de problemas que los chips actuales no manejan bien.
- En el futuro, probablemente veamos sistemas híbridos avanzados donde:





- La CPU coordina tareas.
- La GPU maneja cómputo paralelo.
- Las NPU o TPU hacen inferencia optimizada.
- Los sistemas cuánticos, fotónicos o neuromórficos se activan para tareas no lineales, probabilísticas o ultraeficientes.



*Tecnologías emergentes con potencial disruptivo*

## 2.2 Computación cuántica

La computación cuántica representa un cambio radical en la forma de procesar la información, basado en los principios de la física cuántica en lugar de la lógica binaria tradicional. Mientras que los ordenadores clásicos operan con bits que solo pueden representar un 0 o un 1 en cada momento, los **qubits** (quantum bits) pueden representar simultáneamente ambos estados gracias a la propiedad de **superposición cuántica**.

Además, los qubits pueden estar entrelazados mediante un fenómeno conocido como entrelazamiento cuántico, que permite que el estado de un qubit influya en el estado de otro, aunque estén físicamente separados. Esta capacidad para procesar múltiples estados al mismo tiempo convierte a los ordenadores cuánticos en herramientas potencialmente muy poderosas para ciertos tipos de cálculos.

### *Aplicaciones clave*

La computación cuántica no está diseñada para sustituir a la computación clásica en tareas generales, sino para abordar problemas que actualmente son intratables incluso para los superordenadores más potentes. Algunas de sus aplicaciones destacadas son:

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



- **Optimización combinatoria:** búsqueda de soluciones óptimas en espacios de soluciones muy amplios, como la planificación de rutas, diseño de redes logísticas o gestión de recursos energéticos.
- **Simulación de sistemas físicos complejos:** en química cuántica y ciencia de materiales, donde los métodos clásicos no logran modelar adecuadamente la interacción entre partículas subatómicas.
- **Criptografía:** tanto en su vertiente ofensiva (capacidad de romper algoritmos criptográficos actuales) como defensiva (generación de claves cuánticamente seguras).
- **Algoritmos probabilísticos y de búsqueda:** aceleración de problemas donde se requiere muestreo, inferencia estadística o cálculos de amplitudes cuánticas (por ejemplo, con el algoritmo de Grover o el de Shor).

### *Limitaciones actuales*

A pesar de su potencial, la computación cuántica aún enfrenta importantes desafíos técnicos que limitan su aplicación práctica:

- **Ruido cuántico:** los qubits son extremadamente sensibles a las perturbaciones externas, lo que introduce errores fácilmente.
- **Criogenia extrema:** para mantener la coherencia cuántica, la mayoría de los sistemas actuales requieren temperaturas cercanas al cero absoluto ( $-273,15\text{ °C}$ ), lo que implica altos costes de operación y mantenimiento.
- **Escalabilidad limitada:** los dispositivos cuánticos actuales tienen pocos qubits útiles, y escalar a sistemas funcionales de cientos o miles de qubits corregidos es un reto activo de investigación.
- **Dificultad de integración:** los algoritmos cuánticos aún no pueden ejecutarse de forma autónoma o generalizada, y dependen de arquitecturas híbridas que combinan computación clásica y cuántica.

### *¿Qué opciones reales hay hoy en computación cuántica? Comparativa de enfoques actuales*

Aunque se hable de “la computación cuántica” en general, en realidad **conviven hoy tres modelos tecnológicos diferentes**, cada uno con sus fortalezas, limitaciones y aplicaciones concretas. Estos enfoques no compiten directamente entre sí, sino que se complementan y reflejan distintas estrategias para acercarse a la promesa de la computación cuántica útil.

#### 1. Computación cuántica universal (IBM, Google, IonQ, Quantinuum)

Este modelo intenta construir ordenadores cuánticos capaces de ejecutar **cualquier algoritmo cuántico teóricamente posible**. Es el equivalente cuántico de un ordenador clásico de propósito general.

- **Funcionamiento:** Usa qubits que se controlan con microondas o láseres. La complejidad está en mantener la coherencia cuántica y evitar errores (algo muy difícil).

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



- **Ventajas:** Son los únicos sistemas que pueden ejecutar algoritmos cuánticos famosos como **Shor (factorización)** o **Grover (búsqueda en base de datos)**.
- **Limitaciones:** Actualmente funcionan con **menos de 100 qubits útiles**, y la mayoría de esos qubits se usan para corregir errores.

**Aplicación real hoy:** investigación académica, pruebas de concepto, desarrollo de software cuántico.

**Lo interesante:** estos sistemas aún no superan a los clásicos en tareas útiles, pero son la base sobre la que se construirá el futuro de la computación cuántica general.

## 2. Recocido cuántico (D-Wave)

En lugar de intentar hacerlo todo, D-Wave se centra solo en **resolver problemas de optimización**, como asignar rutas, horarios, tareas o recursos. No necesita qubits universales, sino una red especializada que simula un “paisaje” de soluciones posibles, buscando la mejor.

- **Funcionamiento:** Utiliza el principio del **túnel cuántico** para “saltar” hacia soluciones óptimas en problemas con muchos mínimos locales.
- **Ventajas:** Es **más estable y escalable**: los ordenadores de D-Wave ya funcionan con más de 5.000 qubits físicos (aunque no son equivalentes a los de IBM).
- **Limitaciones:** Solo resuelve **problemas de optimización combinatoria**, no algoritmos universales. No sirve para criptografía, simulación cuántica general, ni otros campos.

**Aplicación real hoy:** logística, producción, investigación operativa.

**Lo interesante:** Es la única tecnología cuántica que **ya se usa en entornos empresariales reales** con resultados prácticos.

## 3. Plataformas híbridas y simuladores (IBM, Google, Amazon, Microsoft)

Como acceder a un ordenador cuántico real no siempre es viable, los principales fabricantes ofrecen entornos donde los usuarios pueden desarrollar algoritmos cuánticos desde su portátil y probarlos en simuladores o máquinas reales en la nube.

- **Funcionamiento:** El código se escribe en frameworks como Qiskit (IBM), Cirq (Google) o Ocean (D-Wave), y se ejecuta en simuladores clásicos o en hardware cuántico remoto.
- **Ventajas:** Permite aprender, experimentar y desarrollar software cuántico sin necesidad de tener un laboratorio.

**NOTA:** IBM Quantum Experience permite a cualquier persona acceder gratis a ordenadores cuánticos reales desde el navegador.

**Aplicación real hoy:** docencia, formación, desarrollo, pruebas de integración.

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



**Lo interesante:** democratizan el acceso a la computación cuántica, haciendo posible que estudiantes y desarrolladores trabajen con esta tecnología desde ya.

## **Conclusión:**

No hay un único tipo de ordenador cuántico, ni una única forma de aplicarlo. En 2025, **la computación cuántica real ya existe, pero con enfoques diferenciados:**

- Si buscas **resolver problemas concretos de optimización**, hoy D-Wave ya ofrece soluciones reales.
- Si tu interés está en **el futuro de la computación universal**, IBM, Google y otros están construyendo las bases, pero aún no hay ventajas prácticas.
- Si quieres aprender o desarrollar software cuántico, **los simuladores y entornos híbridos** te permiten empezar ahora mismo.

\*Para saber más sobre computación cuántica ver la píldora de conocimiento Computación Cuántica básica

## **2.3. Procesadores fotónicos**

### **Introducción**

Los procesadores fotónicos son una tecnología emergente que utiliza luz (fotones) en lugar de electrones para realizar operaciones de cómputo. Sustituyen los transistores electrónicos tradicionales por guías de luz y redes interferométricas, permitiendo realizar cálculos mediante el control y manipulación de señales ópticas.

#### **2.3.1 Principio de funcionamiento**

Estos procesadores emplean componentes ópticos integrados, entre ellos:

- Guías de onda: estructuras que canalizan la luz dentro del chip.
- Interferómetros: dispositivos que utilizan la interferencia de ondas de luz para ejecutar operaciones matemáticas como sumas y multiplicaciones. Encargados
- Moduladores ópticos: regulan la intensidad y fase de la luz.
- Detectores: convierten las señales ópticas en señales eléctricas para interactuar con sistemas electrónicos convencionales.

### **Funcionamiento del Si Chip**

El **Si Chip** es **principalmente óptico**, pero **no es completamente óptico en todos sus componentes**

Aunque el Si chip está diseñado para **procesar señales de luz (fotones)**, en la práctica:



- La **transmisión y manipulación principal de la información es óptica**, mediante guías de onda, interferómetros, moduladores y acopladores.
- Pero el **control de esos elementos se realiza electrónicamente**, mediante circuitos eléctricos integrados que regulan voltajes, temperaturas o campos eléctricos sobre los componentes ópticos.

## Combinan óptica y electrónica

Dentro del chip de silicio fotónico, las partes estrictamente ópticas incluyen:

Guías de onda de silicio: transportan la luz.

- **Interferómetros (Mach-Zehnder)**: realizan operaciones mediante interferencia de luz.
- **Divisores y combinadores de haz**: para dividir o fusionar señales ópticas.
- **Moduladores ópticos**: para codificar señales o controlar la fase.

Estas estructuras están integradas sobre el silicio y operan con luz infrarroja, normalmente en longitudes de onda de telecomunicaciones ( $\sim 1550$  nm).

## Parte electrónica

- **Phase shifters (moduladores de fase)**: controlados mediante corriente eléctrica o calor.
- **Resistencias calefactoras**: utilizadas para cambiar el índice de refracción por efecto térmico.
- **Controladores CMOS**: que gestionan el enrutamiento de señales, ajustes de fase y sincronización.

Es decir, se inyecta energía eléctrica para configurar o ajustar el comportamiento óptico del chip.

### 2.3.2. ¿Qué es un interferómetro?

Un **interferómetro** es un dispositivo óptico que divide un haz de luz en dos o más trayectorias, las recombina después, y genera un patrón de **interferencia**. Este patrón depende de las diferencias de fase entre las ondas que viajan por distintos caminos.. Al hacerlo, la interferencia entre las dos señales de luz depende de:

- Su fase relativa (si las ondas están sincronizadas o desfasadas).
- Su amplitud.

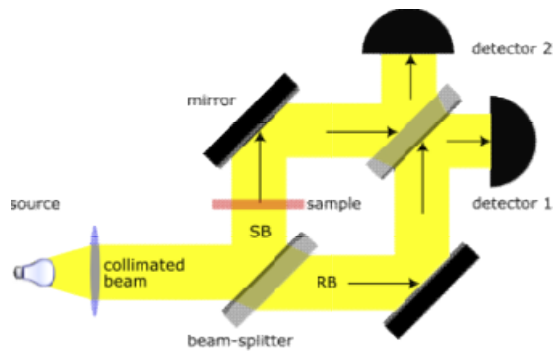
Esta interferencia **puede ser constructiva (se suman) o destructiva (se cancelan)**, y al analizar el resultado en las salidas se obtiene una transformación matemática de la entrada.

### *Interferómetro Mach-Zehnder (MZI): el más común*

Este es el interferómetro que se utiliza en chips fotónicos para computación.

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



## Estructura básica:

- Entrada de luz.
- Divisor de haz (Beam Splitter): divide la luz en dos caminos (superior e inferior).
- Modulador de fase (Phase Shifter): cambia la fase de uno o ambos caminos.
- Combinador (otro Beam Splitter): las dos señales se vuelven a juntar.
- Salidas: dos puertos por los que la luz sale, con diferentes intensidades según la interferencia.

### 2.3.3. Qué operación matemática realiza?

Los interferómetros no hacen “sumas o restas” de números directamente, sino que realizan operaciones sobre **ondas** (funciones matemáticas). Aquí tienes tres ejemplos clave:

#### 1. Suma y resta de ondas (superposición)

Cuando dos ondas se encuentran en un punto, se **suman**:

- Si están en **fase**, se refuerzan (interferencia constructiva).
- Si están en **oposición de fase**, se cancelan (interferencia destructiva).

**Esto es una suma o resta de funciones sinusoidales:**

$$E_{\text{total}}(t) = E_1(t) + E_2(t)$$

#### 2. Medición de diferencias (operación de resta de fases)

Los interferómetros miden **diferencias de fase** entre dos ondas. Esto equivale a medir:

$$\Delta\phi = \phi_1 - \phi_2$$

Y como la fase está relacionada con la distancia recorrida, permite calcular longitudes, índices de refracción, etc., con muchísima precisión.





### 3. Transformadas de Fourier (procesamiento óptico)

Ciertos interferómetros, como el **interferómetro de Fourier (o espectrómetro de Michelson)**, pueden realizar una **transformada de Fourier** sobre una señal:

- Se recoge la interferencia producida al variar la longitud del camino óptico.
- Esta interferencia codifica las frecuencias de la señal original.
- Al aplicar una transformada matemática, se obtiene el **espectro** de frecuencias.

Esto es muy usado en **espectroscopia** y análisis de señales.

#### Ejemplo sencillo: Interferómetro de Michelson

1. Un haz de luz se divide en dos.
2. Cada haz recorre un camino distinto y rebota en un espejo.
3. Al recombinarse, las ondas interfieren.
4. Si uno de los caminos se alarga, el patrón de interferencia cambia.

Ese cambio es directamente proporcional a la **diferencia de longitudes**, y puedes usarlo para:

- Medir desplazamientos pequeñísimos.
- Calcular velocidades (efecto Doppler).
- Detectar ondas gravitacionales (como en LIGO).

#### 2.3.4. Control

El Si Chip es principalmente óptico, pero no es completamente óptico en todos sus componentes

Aunque el Si chip está diseñado para **procesar señales de luz (fotones)**, en la práctica:

- **La transmisión y manipulación principal de la información es óptica**, mediante guías de onda, interferómetros, moduladores y acopladores.
- **Pero el control de esos elementos se realiza electrónicamente**, mediante circuitos eléctricos integrados que regulan voltajes, temperaturas o campos eléctricos sobre los componentes ópticos.

#### Combinan óptica y electrónica

Dentro del chip de silicio fotónico, las partes **estrictamente ópticas** incluyen:

- Guías de onda de silicio: transportan la luz.
- **Interferómetros** (Mach-Zehnder): realizan operaciones mediante interferencia de luz.
- **Divisores y combinadores de haz**: para dividir o fusionar señales ópticas.
- **Moduladores ópticos**: para codificar señales o controlar la fase.

Estas estructuras están integradas sobre el silicio y operan con luz infrarroja, normalmente en longitudes de onda de telecomunicaciones (~1550 nm).

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez



## Parte electrónica

- **Phase shifters** (moduladores de fase): controlados mediante corriente eléctrica o calor.
- **Resistencias calefactoras**: utilizadas para cambiar el índice de refracción por efecto térmico.
- **Controladores CMOS**: que gestionan el enrutamiento de señales, ajustes de fase y sincronización.

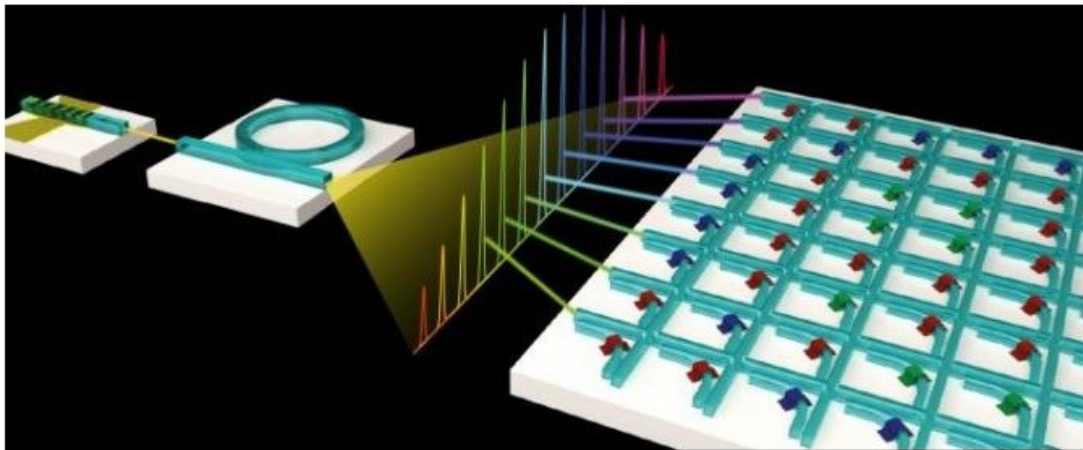
Es decir, se **inyecta energía eléctrica** para configurar o ajustar el comportamiento óptico del chip.

### 2.3.5. Aplicaciones ideales

Los procesadores fotónicos son especialmente eficaces en tareas que requieren operaciones lineales intensivas, como la multiplicación de matrices. Estas operaciones son clave en inteligencia artificial y aprendizaje profundo, especialmente en redes neuronales artificiales.

### Multiplicación de matrices

Se construyen redes de muchos MZIs conectados entre sí, formando una matriz óptica, tal y como podemos ver en la figura:



Representación esquemática de un procesador para multiplicaciones de matrices que funciona con luz. (Foto: WWU/AG Pernice)

La imagen muestra un **procesador fotónico** que utiliza la **interferencia y propagación de luz** para llevar a cabo cálculos matemáticos. En este caso, se trata específicamente de **multiplicaciones de matrices**, que son operaciones fundamentales en:

- Redes neuronales artificiales.
- Machine learning.
- Computación de alto rendimiento.





## ¿Cómo se relaciona con redes neuronales?

En una **red neuronal artificial**, cada capa realiza una operación del tipo:

$$\vec{y} = \sigma(W \cdot \vec{x} + \vec{b})$$

Donde:

- $\vec{x}$  es el vector de entrada
- $W$  es una matriz de pesos
- $\sigma$  es una función de activación

La multiplicación  $W \cdot \vec{x}$  es la parte más costosa computacionalmente, y **es justo lo que hace este procesador óptico**.

## ¿Cómo lo hace con luz?

1. **Se codifican los datos de entrada** (valores del vector o matriz) como intensidades, fases o frecuencias de haces de luz.
2. La luz se propaga a través de una red de guías ópticas (como las líneas azules en la imagen) y pasa por **interferómetros u otros componentes ópticos programables**.
3. Estas estructuras aplican operaciones sobre la luz, equivalentes a multiplicaciones, sumas o transformadas.
4. El resultado (la salida de la multiplicación matricial) se obtiene como una distribución de luz, que puede medirse con fotodetectores.

### 2.3.6. Ventajas principales

- **Alta velocidad:** la luz se propaga más rápido que los electrones, reduciendo la latencia.
- **Bajo consumo energético:** al no generar tanta resistencia ni calor, requieren menos energía.
- **Capacidad de cómputo paralelo:** pueden operar en varias longitudes de onda simultáneamente (multiplexación), permitiendo realizar múltiples operaciones a la vez.
- **Baja latencia:** adecuada para sistemas que necesitan respuestas en tiempo real.

### 2.3.7. Fabricantes principales

- **Lightmatter:** desarrolla procesadores fotónicos como Envisi, integrados en plataformas de computación óptica para centros de datos. Combinan sistemas ópticos y electrónicos para maximizar el rendimiento.
- **Lightelligence:** especializada en aceleradores fotónicos para inteligencia artificial. Han presentado chips como Hummingbird, orientados a realizar operaciones paralelas con alta eficiencia energética.



## 2.3.8. Estado actual del desarrollo

Los procesadores fotónicos se encuentran en una fase avanzada de desarrollo. Aún no han reemplazado a los procesadores electrónicos en aplicaciones generales, pero ya se utilizan como aceleradores externos en clústeres de supercomputación y en tareas específicas de alto rendimiento. Se espera que su adopción aumente conforme avance la tecnología de integración híbrida y el software especializado.

### Retos actuales:

Integración con sistemas electrónicos convencionales: es necesario convertir señales ópticas en eléctricas, lo que introduce cierta complejidad.

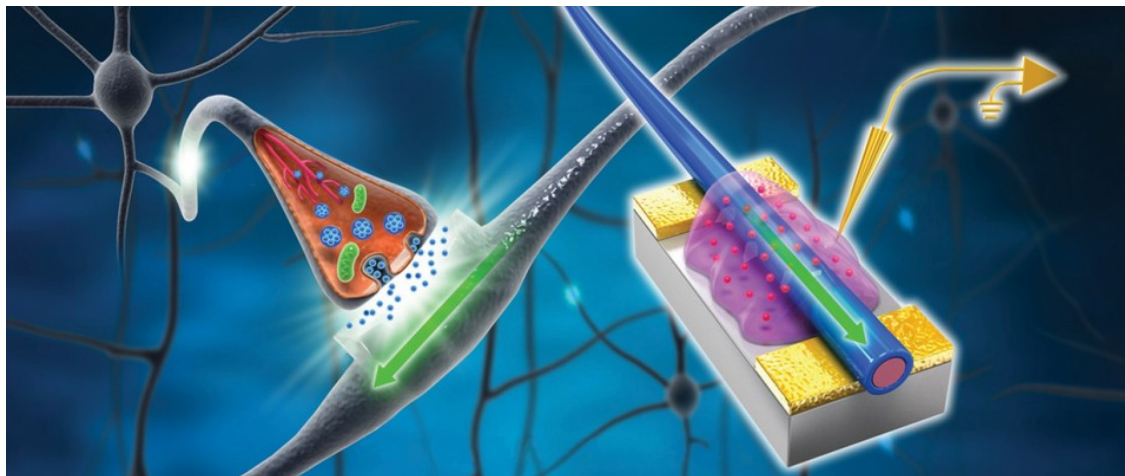
Fabricación a gran escala: los procesos de producción son más complejos y costosos que en la microelectrónica tradicional.

Ecosistema de software: aún se está desarrollando el entorno de programación y las bibliotecas necesarias para aprovechar plenamente esta tecnología.

## 2.4. Chips neuromórficos

### 2.4.1. Definición general

Los chips neuromórficos son dispositivos de hardware diseñados para replicar el funcionamiento del cerebro humano, utilizando una arquitectura inspirada en las redes neuronales biológicas.



CRÉDITO: W. XU ET AL / SCIENCE ADVANCES 2016

A diferencia de los procesadores tradicionales, que ejecutan instrucciones de manera secuencial mediante ciclos de reloj, los chips neuromórficos procesan información de forma asincrónica mediante eventos discretos denominados “spikes”.



Estos eventos simulan los potenciales de acción que utilizan las neuronas reales para comunicarse. La lógica de procesamiento se basa en la activación de neuronas artificiales únicamente cuando se alcanza un determinado umbral de estimulación, lo que permite una computación altamente eficiente en términos energéticos.

- Ref: <https://es.knowablemagazine.org/content/articulo/tecnologia/2022/hacer-que-chips-computadoras-como-neuronas>

## 2.4.2. Fundamentos técnicos

### a. *Arquitectura inspirada en el cerebro*

- Los chips neuromórficos están formados por unidades que simulan el comportamiento de las neuronas y las sinapsis.
- Estas unidades se organizan en redes que operan según el modelo de **Spiking Neural Networks** (SNN), donde la información se transmite mediante picos de voltaje (spikes).
- El procesamiento ocurre de manera distribuida, sin necesidad de sincronización global.

#### \*Lecturas recomendadas:

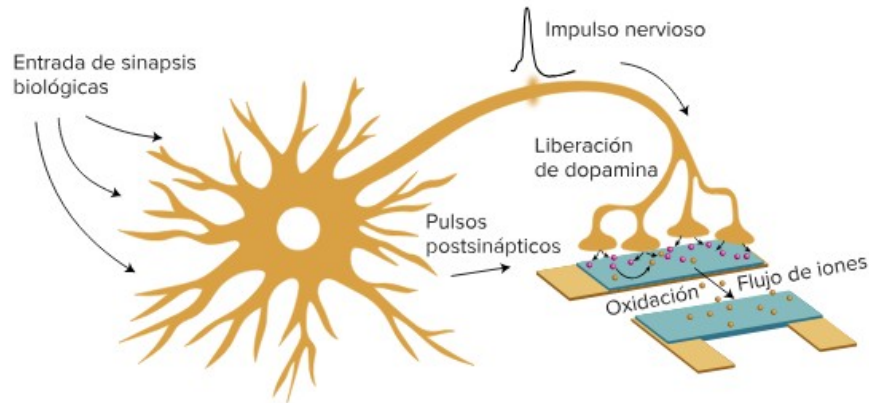
- <https://dev.to/gcjordi/redes-neuronales-espigadas-spiking-neural-networks-y-su-implementacion-en-hardware-neuromorfo-1h5p>
- [https://es.wikipedia.org/wiki/Red\\_neuronal\\_de\\_impulsos](https://es.wikipedia.org/wiki/Red_neuronal_de_impulsos)

### b. *Procesamiento asincrónico y basado en eventos*

- No existe un reloj central que dicte el ritmo de ejecución.
- Cada neurona opera de manera autónoma, activándose solo cuando recibe suficiente estímulo.
- El sistema permanece en reposo mientras no se generan nuevos eventos, lo que reduce significativamente el consumo energético.



## Cómo funcionaría una neurona biohíbrida



FUENTE: ADAPTADO DE S.T. KEENE ET AL / NATURE MATERIALS 2020

REVISTA KNOWABLE

### c. *Aprendizaje local y dinámico*

- Muchos de estos chips implementan mecanismos de aprendizaje inspirados en la neuroplasticidad, como el Spike-Timing Dependent Plasticity (STDP).
- Este tipo de aprendizaje permite la adaptación dinámica del sistema a estímulos nuevos sin requerir un reentrenamiento global.

### 2.4.3. Fabricantes principales y ejemplos relevantes

El desarrollo de chips neuromórficos ha progresado notablemente en los últimos años, destacando por su capacidad para simular grandes redes neuronales con un consumo energético mínimo. Los siguientes fabricantes lideran actualmente este campo:

#### Intel:

Con Loihi 2 y el sistema Hala Point (2024), Intel ha alcanzado la simulación de más de 1.150 millones de neuronas, superando la escala de la corteza cerebral de un primate. Se trata del sistema neuromórfico más grande y avanzado hasta la fecha.

#### IBM

NorthPole integra memoria y procesamiento en un único chip, eliminando el cuello de botella de Von Neumann. En tareas visuales, logra velocidades hasta 22 veces superiores a las GPU actuales, con gran eficiencia energética.

#### BrainChip

Akida permite aprendizaje y ejecución en el borde (edge computing) sin depender de la nube, con aplicaciones en automoción, salud y seguridad. Su arquitectura basada en SNN facilita un procesamiento autónomo y de baja latencia.



## SynSense

Especializada en soluciones de visión y audio en tiempo real con consumo ultra bajo, orientadas a sistemas embebidos, portátiles y autónomos.

## Innatera

Su chip T1 combina procesamiento neuromórfico con aceleración CNN y núcleo RISC-V, permitiendo detección inteligente en entornos de muy bajo consumo, como wearables o dispositivos implantables.

## Qualcomm

En fase experimental, explora la integración de modelos bioinspirados en sistemas móviles para mejorar el procesamiento local y la eficiencia energética en IA embarcada.

### 2.4.4. Aplicaciones típicas

- Sistemas de inteligencia artificial contextual, donde el entorno cambia dinámicamente.
- Robots autónomos que requieren respuestas rápidas y adaptativas.
- Dispositivos con restricciones energéticas severas, como sensores portátiles, implantes médicos, drones y sistemas IoT.
- Reconocimiento de patrones en tiempo real, como la detección de sonidos, imágenes o señales biológicas.

### 2.4.5. Ventajas (resumen)

- **Procesamiento asincrónico:** Aumenta la eficiencia y escalabilidad al no depender de un reloj central.
- **Bajo consumo energético:** Ideal para dispositivos con autonomía limitada.
- **Adaptación en tiempo real:** Capacidad de aprendizaje continuo y modificación de sinapsis según la experiencia.
- **Alta paralelización:** Las neuronas operan simultáneamente sin necesidad de coordinación central.

### 2.4.6. Limitaciones

- **Dificultad de integración con software convencional:** Las herramientas de programación estándar (como TensorFlow o PyTorch) están diseñadas para arquitecturas tradicionales (CPU/GPU).
- **Necesidad de nuevos paradigmas de entrenamiento:** Las técnicas clásicas de entrenamiento supervisado no se adaptan fácilmente a las redes de tipo spiking. Se requieren algoritmos bioinspirados o enfoques de aprendizaje no supervisado.



- **Desafíos de programación:** La lógica de programación es distinta a la computación secuencial, lo que implica una curva de aprendizaje elevada.

## 2.4.7. Convivencia con arquitecturas clásicas

Los chips neuromórficos no están destinados a reemplazar completamente a las CPU o GPU, sino a complementar su funcionamiento. Se pueden utilizar como una capa de procesamiento adaptable dentro de un sistema híbrido, por ejemplo, gestionando tareas perceptivas o reactivas mientras el procesamiento de alto nivel se delega a arquitecturas más tradicionales.

## 3. Redes de Ultra Baja Latencia

### 3.1. Introducción

En los entornos de computación de alto rendimiento (CAR), el tiempo de acceso a los datos y la rapidez en la comunicación entre componentes del sistema son determinantes para el rendimiento global. Las **redes de ultra baja latencia** han surgido como **solución a los cuellos de botella** existentes entre CPU, memoria, dispositivos de almacenamiento y redes. Este tipo de redes permite tiempos de respuesta extremadamente bajos, fundamentales en escenarios como:

- Simulaciones científicas distribuidas
- Modelos de IA de gran escala
- Computación financiera
- Centros de datos hiperconectados
- Sistemas cuánticos híbridos

### 3.2. Tecnologías Emergentes

#### 3.2.1. CXL (Compute Express Link)

**Compute Express Link (CXL)** es un estándar abierto de interconexión de alta velocidad y baja latencia, basado en PCIe (a partir de PCIe 5.0), que permite una comunicación coherente entre CPU, GPU, FPGA y otros aceleradores.

#### Características clave:

##### *Memoria compartida y coherente*

Una de las principales aportaciones de CXL es permitir que CPU, GPU y otros aceleradores accedan a una **zona común de memoria** sin duplicar datos ni generar inconsistencias.

Asignatura: Computación de Alto Rendimiento (CAR)

Profesor: Ricardo Moreno Rodríguez





Todos los dispositivos ven los mismos datos actualizados en tiempo real, lo que simplifica la colaboración entre procesadores y mejora notablemente el rendimiento.

### *Tres protocolos integrados*

CXL se organiza en tres subprotocolos que actúan de forma complementaria:

- **CXL.io**, basado en PCIe, gestiona las funciones básicas de configuración y control.
- **CXL.cache** permite acceder directamente a la caché de otro dispositivo, como si fuera propia.
- **CXL.mem** ofrece acceso remoto a la memoria principal de otro componente, manteniendo la coherencia.

### *Baja latencia y eficiencia energética*

Al manejar la coherencia directamente desde el hardware, se eliminan muchas de las tareas que normalmente recaen en el software. Esto no solo reduce la latencia de acceso a los datos, sino que también disminuye el consumo energético, haciendo que el sistema sea más eficiente y escalable.

### *Pensado para sistemas heterogéneos*

CXL facilita el diseño de arquitecturas donde la memoria puede estar separada físicamente del procesador, algo muy útil en centros de datos y sistemas de computación avanzada, donde distintos tipos de aceleradores deben compartir recursos con rapidez y coherencia.

## 3.2.2. Infiniband HDR (High Data Rate)

**Infiniband** es una arquitectura de red de alta velocidad especialmente diseñada para entornos de supercomputación. Su versión HDR (High Data Rate) representa una evolución significativa en términos de ancho de banda y latencia, posicionándose como una de las tecnologías de interconexión más eficientes y estables en sistemas HPC actuales.

### *Características técnicas destacadas:*

- **Ancho de banda por puerto de hasta 200 Gbps**, escalable mediante la agregación de múltiples lanes físicos.
- **Latencias inferiores a 1 microsegundo**, lo que lo hace ideal para aplicaciones sensibles al retardo, como simulaciones distribuidas o algoritmos de sincronización masiva.
- **Soporte para RDMA (Remote Direct Memory Access)**, que permite a un nodo acceder directamente a la memoria de otro sin pasar por la CPU ni por el sistema operativo. Esta capacidad reduce significativamente la sobrecarga de procesamiento y mejora la eficiencia global del sistema.
- **Topologías escalables** como torus, fat-tree o dragonfly, utilizadas en arquitecturas de clúster que requieren eficiencia en el enrutamiento de datos y tolerancia a fallos.



## *Aplicación en sistemas reales:*

Tecnologías como HDR están actualmente implementadas en algunos de los **supercomputadores más potentes del mundo**, como:

- **Summit** (Oak Ridge National Laboratory, EE.UU.)
- **Perlmutter** (Lawrence Berkeley National Lab)

Ambos sistemas integran miles de nodos heterogéneos con GPUs, donde una red de interconexión ultrarrápida es fundamental para coordinar el cómputo paralelo sin crear cuellos de botella.

## *Ventajas operativas:*

- Reducción significativa del **jitter** (variación de latencia), fundamental en algoritmos con sincronización fina.
- **Menor consumo energético** en comparación con soluciones Ethernet equivalentes, especialmente en entornos de carga continua.
- **Alta eficiencia en la transmisión de mensajes cortos**, aspecto crítico en Computación de Alto Rendimiento (CAR), donde los procesos suelen comunicarse con alta frecuencia usando pequeños paquetes de datos.

## *Entorno de uso:*

Infiniband HDR se está consolidando como la solución preferida en:

- Centros de datos HPC con arquitecturas **híbridas** (CPU + aceleradores).
- Plataformas para simulación científica, aprendizaje profundo distribuido y modelado de sistemas físicos complejos.

### 3.2.3. Quantum Networking

Aunque todavía en fases iniciales de implementación práctica, las **redes cuánticas** apuntan a redefinir la forma en que concebimos la comunicación entre sistemas, especialmente en lo relativo a la latencia y la seguridad.

## *Fundamentos científicos:*

A diferencia de las redes tradicionales, estas se basan en **qubits** (unidades de información cuántica) y en principios como:

- **Entrelazamiento cuántico:** permite que dos partículas compartan un estado común, de modo que cualquier cambio en una afecta inmediatamente a la otra, sin importar la distancia.
- **Teleportación cuántica:** mecanismo mediante el cual el estado de un qubit se transfiere a otro distante, utilizando una combinación de entrelazamiento y comunicación clásica.





Cabe aclarar que **no se transmite información clásica instantáneamente**, pero se posibilitan operaciones con latencia efectivamente nula desde la perspectiva del observador, lo que tiene implicaciones profundas en el diseño de algoritmos híbridos.

#### *Aplicaciones previstas:*

- **Criptografía cuántica (QKD)**: comunicaciones imposibles de interceptar sin ser detectadas.
- **Interconexión de procesadores cuánticos distribuidos**, un paso esencial hacia redes de computación cuántica a gran escala.
- **Reducción de pasos clásicos en algoritmos híbridos cuántico-clásicos**, acelerando tareas críticas como el entrenamiento de modelos en sistemas cuánticos.

#### *Limitaciones actuales:*

- Las **distancias alcanzables** son aún muy limitadas (del orden de decenas a cientos de kilómetros).
- Alta **sensibilidad a la decoherencia**: los qubits pueden perder su estado si interactúan con el entorno.
- Requieren de **repetidores cuánticos** para ampliar el alcance, tecnología que actualmente se encuentra en fase experimental.

#### *Entorno de desarrollo:*

Actualmente, redes cuánticas experimentales han sido implementadas en:

- **Países Bajos** (Red cuántica nacional entre laboratorios).
- **China** (enlace cuántico de más de 1.200 km con estaciones intermedias).
- Proyectos europeos como **EuroQCI (Quantum Communication Infrastructure)**, que buscan crear redes cuánticas soberanas de alcance continental.

## 4. Software inteligente y autoadaptativo

### 4.1. Introducción

La Computación de Alto Rendimiento (CAR) avanza hacia una nueva era caracterizada por sistemas exa- y zettaescala. Este salto conlleva desafíos técnicos de una magnitud sin precedentes: millones de núcleos, heterogeneidad de hardware, tasas de fallo elevadas y demandas energéticas críticas. Frente a esta complejidad, el software tradicional resulta insuficiente.

En este contexto, surge la necesidad de soluciones inteligentes, capaces de adaptarse dinámicamente a un entorno cambiante y exigente. El software autoadaptativo representa una de las respuestas más prometedoras a estos retos, permitiendo una gestión autónoma, eficiente y resiliente de los recursos computacionales.



## 4.2. ¿Qué es el software autoadaptativo?

El software autoadaptativo es aquel que puede modificar su comportamiento de forma autónoma durante su ejecución, en función de variables como la carga de trabajo, el estado del sistema o el entorno de ejecución.

Esta capacidad no se limita a simples reacciones predefinidas, sino que incorpora mecanismos de aprendizaje automático, análisis predictivo y reconfiguración dinámica para optimizar su funcionamiento en tiempo real.

Ejemplos actuales incluyen gestores de recursos que adaptan su planificación según el uso histórico o sistemas que modifican parámetros internos sin intervención humana.

## 4.3. Sistemas de gestión inteligentes

En los entornos HPC modernos, los sistemas de gestión inteligente cumplen un papel esencial. Estos sistemas no solo monitorizan el estado del hardware y las aplicaciones, sino que pueden redistribuir la carga entre nodos, migrar procesos en respuesta a errores o modificar dinámicamente la configuración del sistema. Por ejemplo, en el supercomputador Aurora del Argonne National Laboratory, se están integrando soluciones capaces de ajustar la frecuencia de procesadores y gestionar fallos de memoria sin intervención externa, manteniendo la operación continua del sistema.

## 4.4. Técnicas empleadas

Diversas disciplinas confluyen en el desarrollo de software autoadaptativo. El uso de modelos predictivos permite anticipar fallos o sobrecargas, mientras que el control adaptativo garantiza que variables críticas, como temperatura o rendimiento, se mantengan dentro de márgenes seguros.

En paralelo, técnicas como la metaprogramación o la infraestructura definida por software (SDI) facilitan la reconfiguración del sistema a nivel lógico sin necesidad de reinicios o reprogramación manual.

## 4.5. Importancia en sistemas exa- y zetta-escala

A medida que se desarrollan sistemas con más de  $10^{18}$  operaciones por segundo, la fiabilidad y eficiencia energética se convierten en prioridades absolutas. Se estima que un sistema exaescala podría experimentar fallos de hardware cada pocas horas. Por ello, es imprescindible contar con software que no solo detecte los fallos, sino que los mitigue o repare automáticamente. Además, la gestión energética en tiempo real, ajustando la actividad de los nodos a la carga actual, es fundamental para cumplir con los límites presupuestarios y de sostenibilidad de los grandes centros de datos.



## 4.6. Ejemplos actuales y tendencias

Algunos ejemplos ya se encuentran en fase de producción o prototipado avanzado. Intel DAOS, un sistema de almacenamiento optimizado para exaescala, adapta su comportamiento según los patrones de acceso observados.

Por otro lado, gestores de colas como SLURM están siendo enriquecidos con módulos de inteligencia artificial capaces de anticipar la mejor distribución de tareas en clústeres complejos.

También destacan los enfoques de computación autoconsciente, en los que el software mantiene un modelo interno de su comportamiento y entorno para tomar decisiones más eficaces a medio y largo plazo.

## 4.7. Conclusión

El software inteligente y autoadaptativo no es una opción, sino una necesidad estructural para la computación a gran escala. Su desarrollo, validación e implementación representan un eje estratégico para garantizar que los futuros sistemas exa- y zettaescala no solo sean potentes, sino también sostenibles, robustos y operables sin intervención constante.

En los próximos años, se espera que este tipo de software no solo gestione recursos, sino que también se convierta en un agente decisor dentro del sistema, capaz de colaborar con usuarios y otros sistemas autónomos en la toma de decisiones técnicas complejas.



## Parte 2. Debate aplicado y crítico

---

### Procesadores híbridos (CPU + GPU + IA)

- ¿Qué ventajas aporta realmente integrar CPU, GPU y NPU en un mismo chip frente a la clásica separación de tareas?
- ¿Cómo influye la memoria jerárquica compartida en la eficiencia energética en entornos reales como móviles, estaciones de trabajo o centros de datos?
- ¿Qué dilemas éticos o de gobernanza plantea el hecho de que una IA embebida tome decisiones autónomas sin pasar por la nube ni por sistemas auditables?

### Supercomputadores heterogéneos

- ¿Qué tipo de problemas reales justifican la integración de arquitecturas emergentes como cuántica, fotónica o neuromórfica?
- Si hoy tuvieras que diseñar un clúster heterogéneo para resolver un problema concreto (ej. predicción climática, simulación de proteínas, optimización logística), ¿cómo lo distribuirías?
- ¿Qué limitaciones prácticas (coste, mantenimiento, software, personal cualificado) impiden una adopción más rápida de estas tecnologías?

### Redes de ultra baja latencia

- ¿En qué escenarios concretos se vuelve crítica la latencia por debajo del microsegundo? ¿Puedes pensar en un ejemplo donde la latencia no sea negociable?
- ¿Qué consecuencias tendría elegir CXL en lugar de Infiniband HDR o viceversa para diseñar un sistema orientado a IA generativa distribuida?
- Si la red cuántica no transmite información clásica instantáneamente, ¿por qué genera tanto interés? ¿Es una disrupción o una exageración?

### Co-diseño HPC + IA

- ¿Qué riesgos implica delegar parte del diseño arquitectónico a algoritmos de IA? ¿Quién asume la responsabilidad en caso de fallo?
- ¿Podría la IA sustituir a los ingenieros de arquitectura en algunos aspectos del diseño de sistemas HPC? ¿Hasta qué punto?
- ¿Qué ventajas aporta el co-diseño frente a arquitecturas evolucionadas por capas? ¿Conoces algún caso donde se haya aplicado con éxito?

### Software inteligente y autoadaptativo

- ¿Deberíamos confiar decisiones críticas (energía, errores, rendimiento) a un sistema completamente autónomo en un centro de datos?



Universitat d'Alacant  
Universidad de Alicante

- ¿Qué medidas de control o supervisión deben acompañar a este tipo de software para evitar decisiones no deseadas?
- ¿Ves factible un futuro donde el administrador de sistemas actúe como “entrenador” de un software autoconsciente más que como operador técnico?



## Bibliografía y enlaces recomendados

### 2.4 Chips neuromórficos

- Knowable Magazine (2022). 'Hacer que los chips piensen como neuronas'. <https://es.knowablemagazine.org/content/articulo/tecnologia/2022/hacer-que-chips-computadoras-como-neuronas>
- Jordi, G. C. (dev.to). 'Redes Neuronales Espigadas y su implementación en hardware neuromórfico'. <https://dev.to/gcjordi/redes-neuronales-espigadas-spiking-neural-networks-y-su-implementacion-en-hardware-neuromorfico-1h5p>
- Wikipedia. 'Red neuronal de impulsos'. [https://es.wikipedia.org/wiki/Red\\_neuronal\\_de\\_impulsos](https://es.wikipedia.org/wiki/Red_neuronal_de_impulsos)
- Intel Newsroom (2024). 'Hala Point'. <https://newsroom.intel.com/es/nuevas-tecnologias/intel-newsroom-archivo-2024>
- Wikipedia. 'Computadora cognitiva'. [https://es.wikipedia.org/wiki/Computadora\\_cognitiva](https://es.wikipedia.org/wiki/Computadora_cognitiva)
- Wikipedia. 'BrainChip'. <https://en.wikipedia.org/wiki/BrainChip>
- Global Growth Insights. 'Neuromorphic Chip Market'. <https://www.globalgrowthinsights.com/blog/neuromorphic-chip-market-49>
- FirstIgnite. 'Neuromorphic Computing Advancements in 2024'. <https://www.firstignite.com/es/exploring-the-latest-neuromorphic-computing-advancements-in-2024>

### 3.2 Tecnologías emergentes

- Compute Express Link (CXL) Specification: <https://www.computeexpresslink.org/>
- Wikipedia. 'Infiniband'. <https://es.wikipedia.org/wiki/InfiniBand>
- Oak Ridge National Laboratory. 'Summit Supercomputer'. <https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit/>
- Lawrence Berkeley National Lab. 'Perlmutter Supercomputer'. <https://crd.lbl.gov/divisions/hpcf/facilities/perlmutter/>
- EuroQCI (Quantum Communication Infrastructure): <https://digital-strategy.ec.europa.eu/en/policies/quantum-communication-infrastructure>



Universitat d'Alacant  
Universidad de Alicante

#### 4. Software autoadaptativo

- Intel DAOS Project. <https://www.intel.com/content/www/us/en/high-performance-computing/daos-storage-overview.html>
- SLURM Workload Manager. <https://slurm.schedmd.com/>
- Argonne National Laboratory. 'Aurora Exascale Supercomputer'. <https://www.alcf.anl.gov/alcf-resources/aurora>
-