

# Tema 01: Introducción a la Arquitectura de Computadoras

**Arquitectura de Computadoras**

**Ing. Nicolás Majorel Padilla ([npadilla@herrera.unt.edu.ar](mailto:npadilla@herrera.unt.edu.ar))**

<http://microprocesadores.unt.edu.ar/arqcom/>

# Temas que veremos

---

- ▶ Conceptos básicos.
  - ▶ ¿Qué es una computadora?
  - ▶ Criterios fundamentales de decisión.
  - ▶ Partes de una computadora.
    - ▶ Diferencias fundamentales.
- ▶ ¿Qué es Arquitectura de Computadoras?
  - ▶ Modelos e influencias.
- ▶ Grandes ideas en Arquitectura de Computadoras.

# Lectura recomendada

---

- ▶ Computer Organization and Design, RISC-V Edition (2da ed.)
  - ▶ Sección 1.1: *Introduction*
  - ▶ Sección 1.2: *Seven Great Ideas in Computer Architecture*
  - ▶ Sección 1.3: *Below Your Program*

# ¿Qué es una computadora?

- ▶ Tradicionalmente se conocen 4 tipos principales:
  - ▶ Computadoras personales.
  - ▶ Servidores.
  - ▶ Supercomputadoras.
  - ▶ Sistemas Embebidos.
- ▶ Las diferencias están dadas básicamente por tres criterios fundamentales: Costo, Performance y Consumo de Energía.
- ▶ Sin embargo, surgieron dudas en los últimos años.
  - ▶ No es lo mismo una PC que una laptop.
  - ▶ Un dispositivo móvil (celular), *¿es una PC o sistema embebido?*
  - ▶ *¿Qué sería un vehículo autónomo?*
  - ▶ Los nuevos Datacenters para IA, *¿son servidores o supercomputadoras?*

# ¿Qué es una computadora?

---

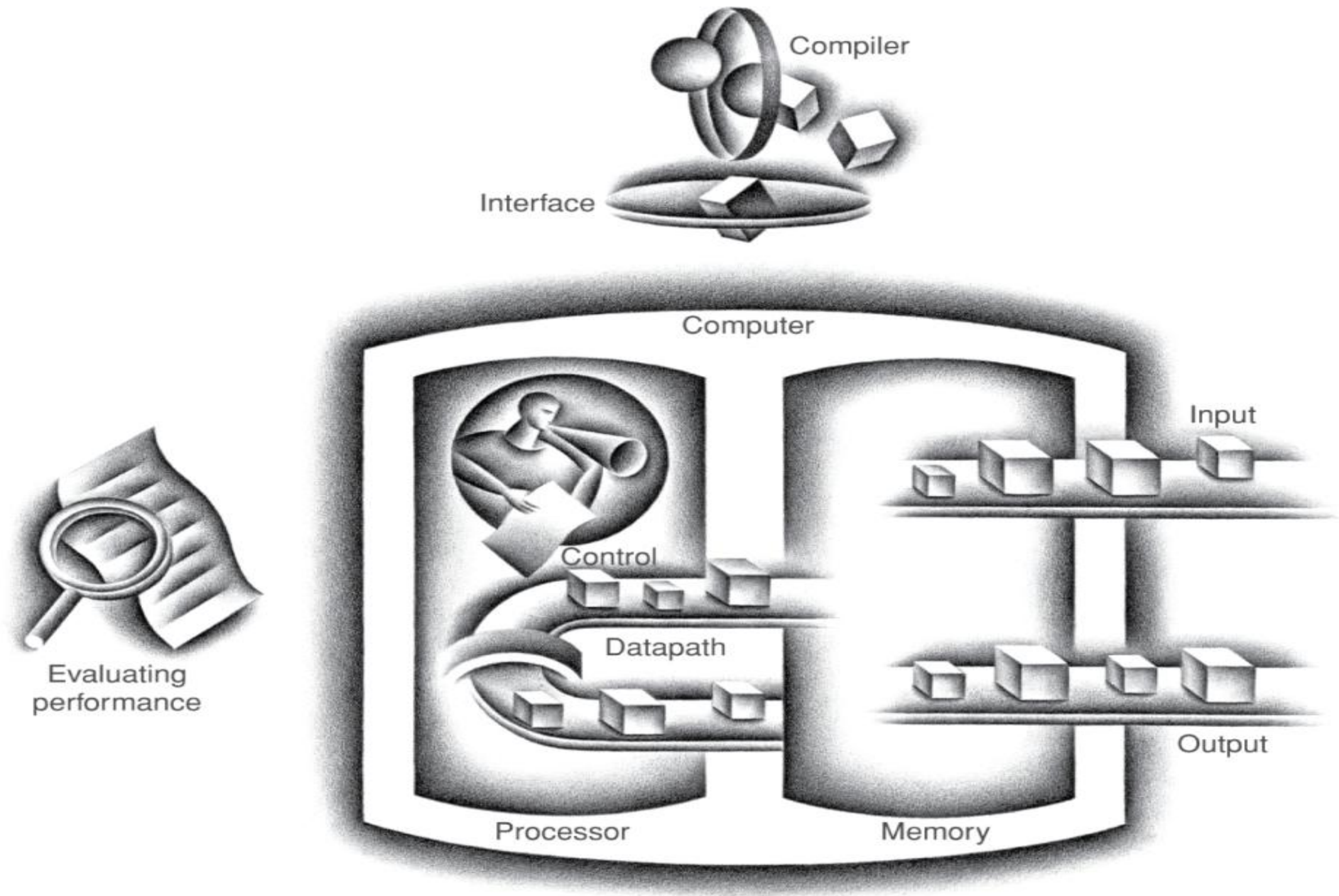
- ▶ Una clasificación más actualizada podría ser la siguiente:
  - ▶ PCs
  - ▶ *Laptops (o notebooks)*
  - ▶ Dispositivos móviles
    - ▶ Y estas tres categorías tienen a su vez subcategorías: gama de entrada, gama media y tope de gama.
  - ▶ *Servers.*
  - ▶ *Cloud servers (Warehouse Scale Computers)*
  - ▶ Supercomputadoras para HPC (*High Performance Computing*)
    - ▶ Las mejores del mundo están en <http://top500.org>
  - ▶ Datacenters para IA.
  - ▶ Sistemas Embebidos, con un muy amplio espectro.
  - ▶ Computadoras cuánticas.

# ¿Qué es una computadora?

---

- ▶ Los tipos de computadora mencionados son muy diferentes entre sí
  - ▶ *¿Cierto?*
- ▶ Mantienen los mismos tres criterios fundamentales para establecer sus diferencias:
  - ▶ **Costo, Performance y Consumo de Energía.**
  - ▶ Los veremos en el Tema 02.
- ▶ Conceptualmente, todas están compuestas de la misma manera...

# Partes de una computadora



# Partes de una computadora

---

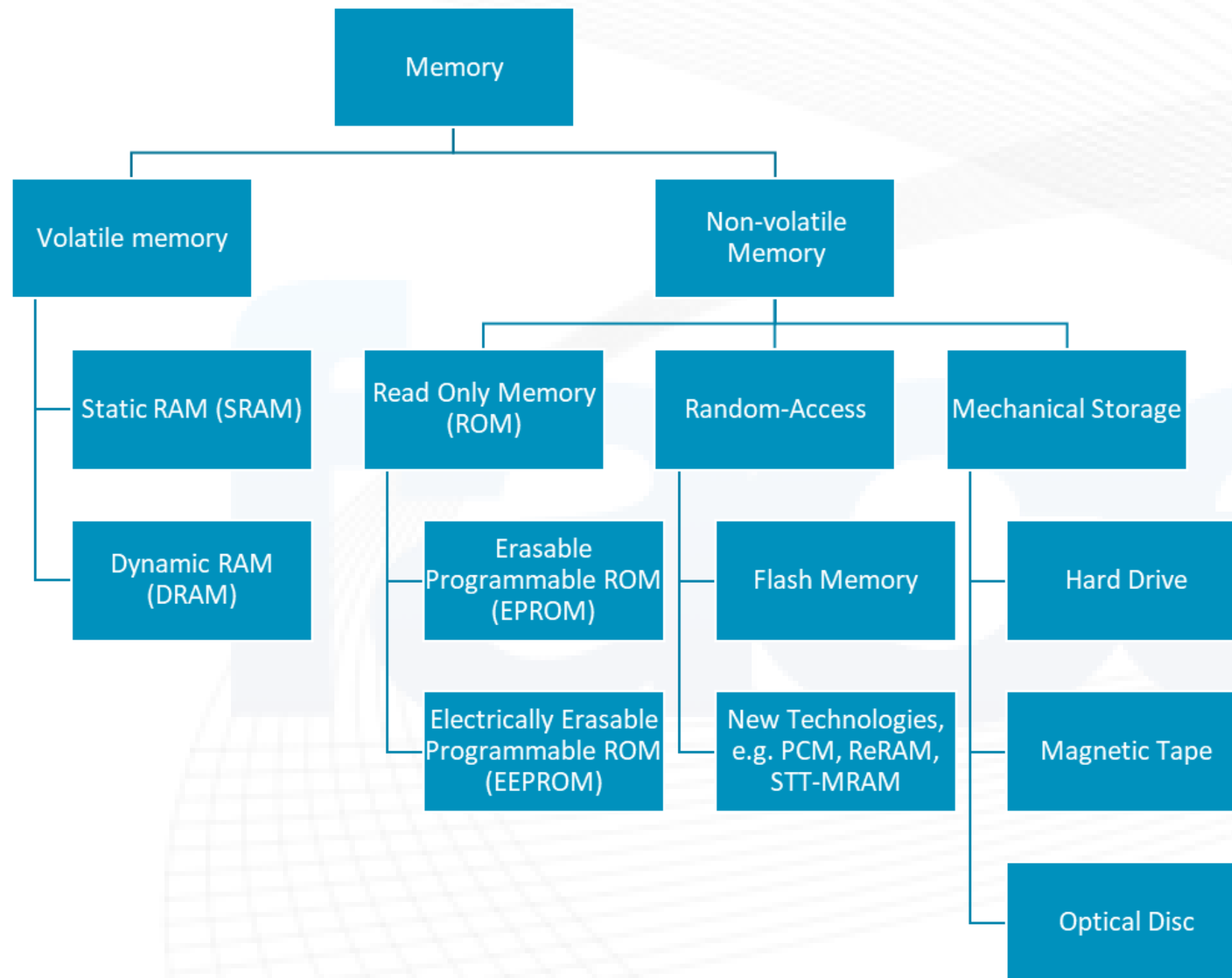
- ▶ Es un sistema **complejo**.
- ▶ Basado en el esquema Von Neumann.
- ▶ 5 componentes principales:
  - ▶ Procesador:
    - ▶ **Camino de datos**, por donde se mueven los datos.
    - ▶ **Control**, maneja y sincroniza al camino de datos.
  - ▶ **Memoria**, que almacena la información.
  - ▶ Dispositivos de **Entrada**, para ingresar información.
  - ▶ Dispositivos de **Salida**, para mostrar información.
  - ▶ Todos interconectados mediante **Buses**.

# Procesador

---

- ▶ Ejecuta los programas.
- ▶ Ciclo de la instrucción
  - ▶ Fetch → Decode → Execution
- ▶ Ejemplo de SMM:
  - ▶ ISA (RV32I)
  - ▶ Camino de datos (Multiciclo)
  - ▶ Control (Microprogramado)
- ▶ **Repasar Temas 03, 05 y 06 de SMM**

# Memoria – Muchos tipos



- ▶ Por modo de acceso
  - ▶ **Primaria:** accede directamente el procesador (RAM).
  - ▶ **Secundaria:** se accede a través de I/O.
- ▶ Nos focalizaremos en otro aspecto...

# Memoria – Objetivos

---

- ▶ El foco no estará necesariamente en las tecnologías de memoria, sino en qué buscamos de ellas.
- ▶ Como primera idea, queremos que la memoria sea **rápida**.
  - ▶ *¿Cuál de las memorias anteriores son las más rápidas?*
- ▶ Luego, queremos que sea **grande**.
  - ▶ *¿Cuál de las memorias anteriores son las más grandes?*
- ▶ Finalmente, sería ideal que sea **barata**, en costo por byte.
  - ▶ *¿Cuál de las memorias anteriores son las más baratas?*
- ▶ No existe (actualmente) una tecnología de memoria que cumpla con los tres requisitos simultáneamente.

# Números que todo ingeniero debe saber

- ▶ En cada ciclo de reloj se hace mínimamente una transferencia entre registros.
- ▶ *¿Cuál es la frecuencia de un microprocesador actual?*
  - ▶ La cantidad de transferencias que pueden hacerse en 1 s es altísima.

|                                             |           |
|---------------------------------------------|-----------|
| Un ciclo de reloj                           | 0,25 ns   |
| Acceso a memoria más rápida                 | 0,4 ns    |
| Acceso a Memoria principal (DRAM)           | 80 ns     |
| Acceso a un disco SSD                       | 50-150 us |
| Acceso a un disco rotacional                | 1-10 ms   |
| Tiempo para tipear una palabra              | 1 s       |
| Tiempo para abrir PowerPoint en mi notebook | 3 s       |

- ▶ ¡¡Acceso a DRAM es **200 veces más lento** que el acceso a la memoria más rápida!!
- ▶ Si el ciclo de reloj fuera de 1 s, un acceso a un SSD demoraría entre 2 y 4 **días**.
- ▶ La diferencia entre un ciclo de reloj y un acceso a Memoria RAM es ENORME.
- ▶ La diferencia entre un ciclo de reloj y cualquier tarea de humanos es ABISMAL.

# Convención de prefijos

- ▶ Nuevos prefijos para potencias de 2
  - ▶ Para evitar ambigüedades.
  - ▶ Dispositivos de I/O con decimal, todo lo demás con binario.

| Decimal term | Abbreviation | Value     | Binary term | Abbreviation | Value     | % Larger |
|--------------|--------------|-----------|-------------|--------------|-----------|----------|
| kilobyte     | KB           | $10^3$    | kibibyte    | KiB          | $2^{10}$  | 2%       |
| megabyte     | MB           | $10^6$    | mebibyte    | MiB          | $2^{20}$  | 5%       |
| gigabyte     | GB           | $10^9$    | gibibyte    | GiB          | $2^{30}$  | 7%       |
| terabyte     | TB           | $10^{12}$ | tebibyte    | TiB          | $2^{40}$  | 10%      |
| petabyte     | PB           | $10^{15}$ | pebibyte    | PiB          | $2^{50}$  | 13%      |
| exabyte      | EB           | $10^{18}$ | exbibyte    | EiB          | $2^{60}$  | 15%      |
| zettabyte    | ZB           | $10^{21}$ | zebibyte    | ZiB          | $2^{70}$  | 18%      |
| yottabyte    | YB           | $10^{24}$ | yobibyte    | YiB          | $2^{80}$  | 21%      |
| ronnabyte    | RB           | $10^{27}$ | robibyte    | RiB          | $2^{90}$  | 24%      |
| queccabyte   | QB           | $10^{30}$ | quebibyte   | QiB          | $2^{100}$ | 27%      |

# Visión general

---

- ▶ Arquitectura de computadoras consiste fundamentalmente en **disimular las enormes diferencias de latencia/ancho de banda** de todas las partes que componen una computadora.
- ▶ Estas diferencias son los principales motivos de muchas técnicas que veremos durante el curso:
  - ▶ Paralelismo a nivel instrucción.
  - ▶ Ejecución fuera de orden.
  - ▶ Ejecución especulativa.
  - ▶ Múltiples niveles de cachés.

# Visión general

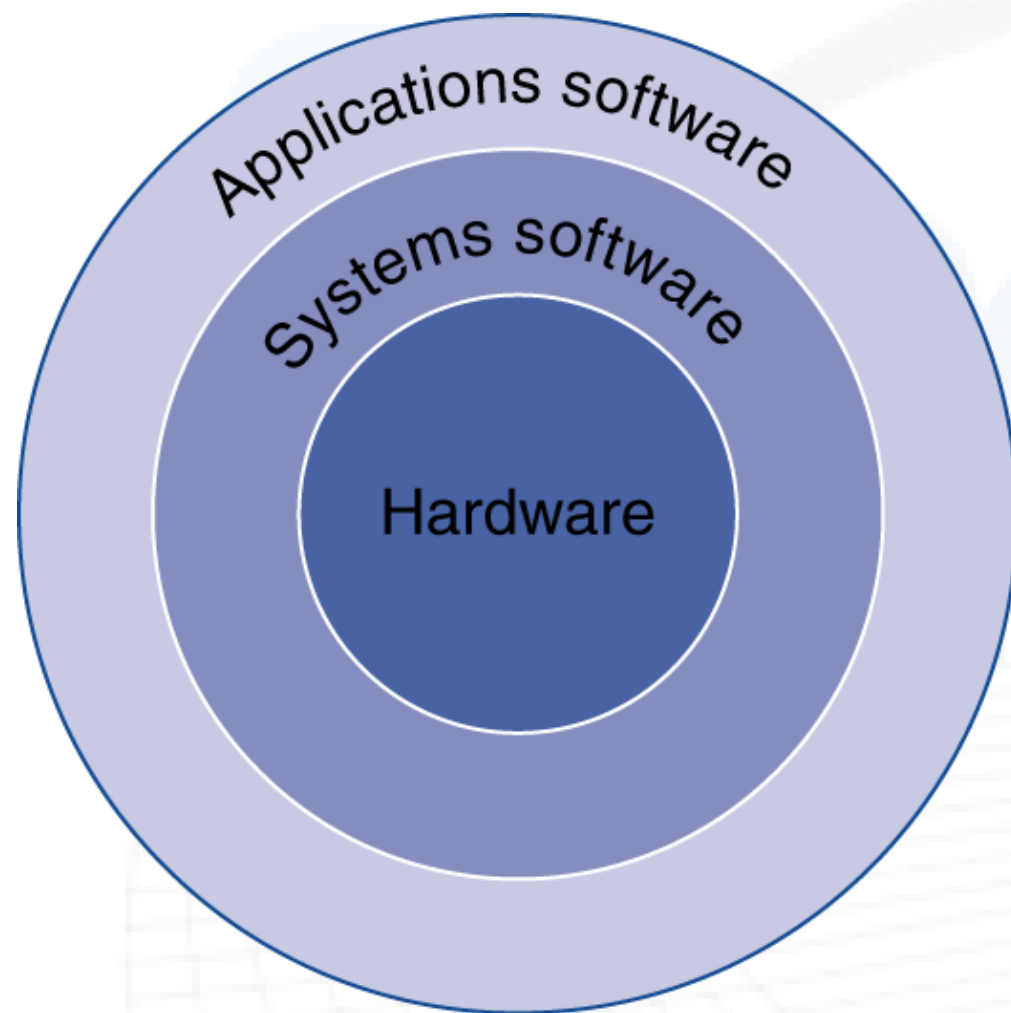
---

- ▶ Arquitectura de computadoras implica resolver desafíos complejos que abarcan el *stack* de tecnología completo.
- ▶ Encarar estos desafíos transversales implica sintetizar perspectivas de alto nivel, con implementaciones de bajo nivel.
- ▶ Requiere un conocimiento profundo de:
  - ▶ Características y comportamiento de la tarea a realizar
  - ▶ Limitaciones de cada característica arquitectural.
  - ▶ Interacciones a nivel sistema.

# Modelo por Capas

---

- ▶ Permite manejar **abstracciones**.
  - ▶ A través de **interfaces** entre ellas.



- ▶ Software de sistema incluye:
  - ▶ Sistema Operativo
  - ▶ Compilador
  - ▶ Enlazador
  - ▶ Linker/Loader
  - ▶ Ensamblador

# Niveles de código

## Levels of Program Code

High Level Language  
(C++, Python, Java)

```
temp = v[k];  
v[k] = v[k+ 1];  
v[k+ 1] = temp;
```

*Compiler*

Assembly Language  
(ARM, MIPS, x86)

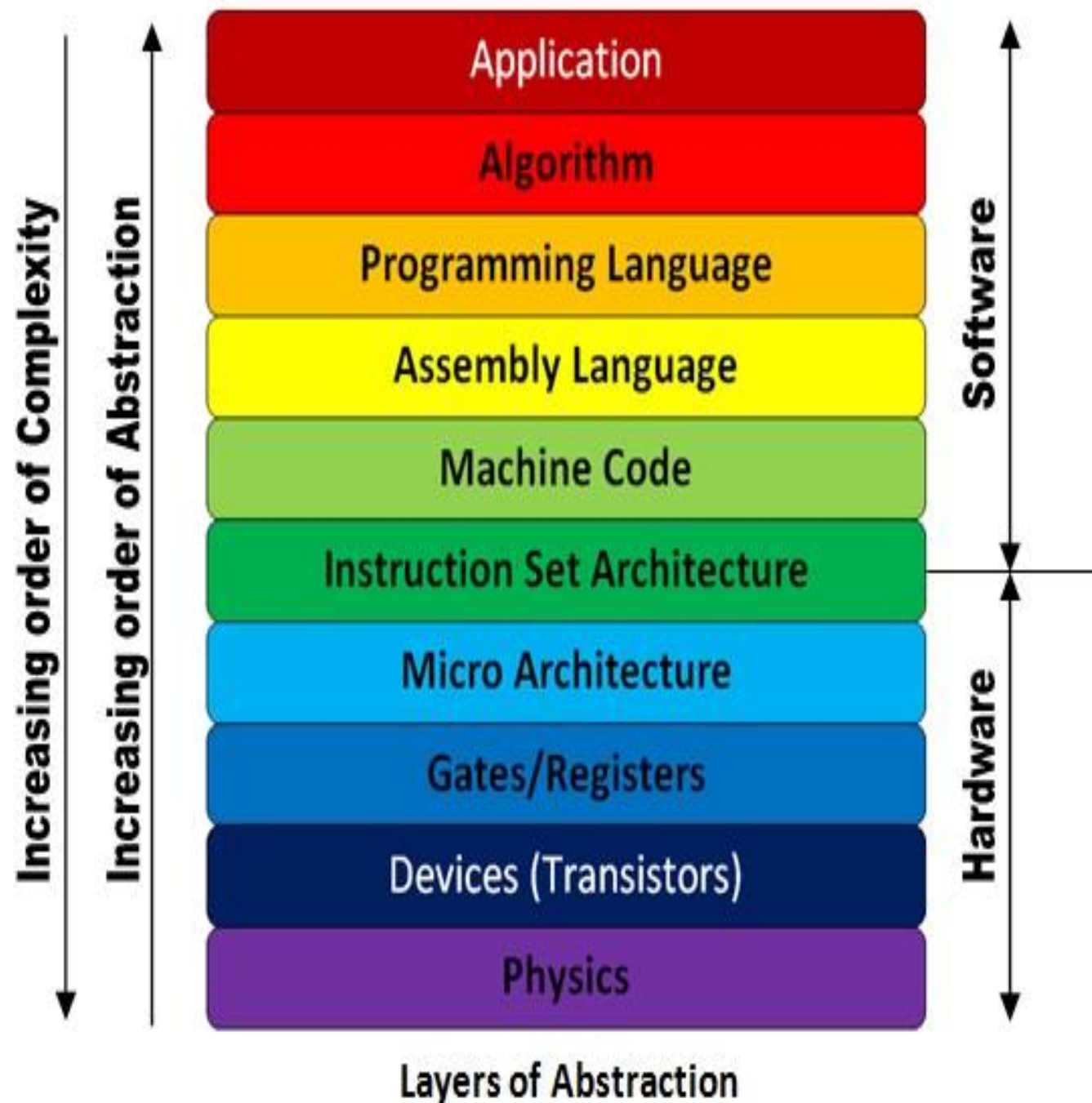
```
lw $t0, 0($2)  
lw $t1, 4($2)  
sw $t1, 0($2)  
sw $t0, 4($2)
```

*Assembler*

Machine Language

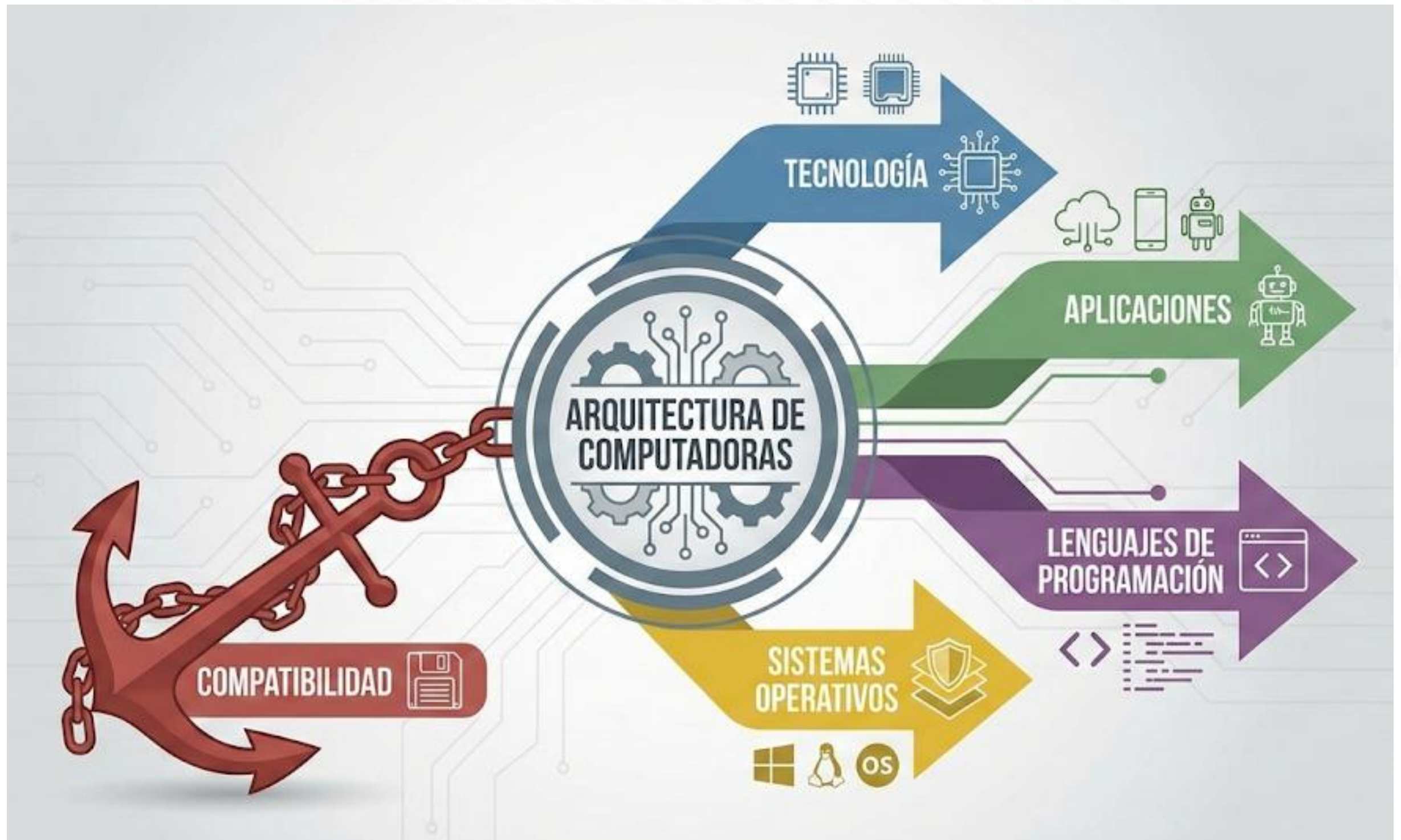
```
0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111
```

# Modelo Jerárquico



- ▶ Arquitectura de computadoras se encarga de la **coordinación de distintos niveles de abstracción.**
  - ▶ Cómo una decisión sobre el Hw afecta a las capas de Sw.
  - ▶Cuál capa es mejor para intentar resolver un problema.

# Influencias en Arquitectura de Comp.



# Influencias en Arquitectura de Comp.

---

- ▶ ¡Fuerzas que cambian permanentemente!
- ▶ No existe una “convergencia” a una solución “óptima”.
- ▶ La mejor solución hoy no necesariamente será la mejor solución mañana.
- ▶ Una idea descartada ayer puede ser volver a ser considerada en el futuro.

# Grandes ideas en Arquitectura

---

- ▶ El concepto de Abstracción mencionado es una de las **Grandes Ideas en Arquitectura de Computadoras**.
- ▶ Son conceptos muy importantes, que veremos varias veces durante toda la materia.
- ▶ Más allá de los cambios tecnológicos, estas ideas fundamentales perduran con el tiempo.

# Grandes ideas en Arquitectura

---

- ▶ **Abstracción** para simplificar los diseños.
  - ▶ Diferentes niveles de representación, en los que los detalles de un nivel inferior están ocultos a los niveles superiores.
- ▶ **Hacer más rápido el caso más común.**
  - ▶ Implica conocer cuál caso es el más común.
- ▶ Mejorar performance mediante el uso de **Paralelismo**.
- ▶ Mejorar performance mediante el uso de **Pipelining**.
- ▶ Mejorar performance mediante el uso de **Predicción**.
  - ▶ Tratar de adivinar el futuro.
  - ▶ “Es mejor pedir perdón que pedir permiso” 😊
- ▶ Usar una **Jerarquía de Memorias**.
  - ▶ Para brindar a los programas la ilusión de que la memoria realmente es muy grande, muy rápida y muy barata.

# Arquitectura de Computadoras en 2026

---

- ▶ Bastante diferente que 20 años atrás.
- ▶ Nuevas tecnologías
  - ▶ El chip más grande de la historia (2019).
  - ▶ Nuevas conexiones.
- ▶ Nuevas necesidades de procesamiento
  - ▶ **¡Estamos en medio de una revolución de la Inteligencia Artificial! (desde 2023)**
  - ▶ Supercomputadoras superaron la barrera de 1 exaFLOP (2022).
  - ▶ Comunicaciones 5G.
- ▶ Nuevos problemas
  - ▶ Brechas de seguridad descubiertas (2018).

# Resumen final

---

- ▶ Hay diferentes tipos principales de computadoras
- ▶ Categorizadas mediante **tres criterios fundamentales**:
  - ▶ **Costo, Performance y Consumo de Energía.**
- ▶ Todas compuestas por **los mismos 5 componentes**.
  - ▶ Procesador (**Camino de Datos y Control**), Memoria, Entrada y Salida.
- ▶ No existe ninguna memoria que sea al mismo tiempo rápida, grande y barata.
- ▶ La memoria principal es muchísimo más lenta que el procesador.
  - ▶ Y los dispositivos de Entrada y Salida... abismalmente peor.
- ▶ Diferencias de **muchos órdenes de magnitud** en los tiempos de las distintas partes de una computadora.

# Resumen final

---

- ▶ **Arquitectura de Computadoras**
  - ▶ Organización para ‘disimular’ las enormes diferencias de ancho de banda de los distintos componentes.
  - ▶ Coordinación de distintos niveles de abstracción
  - ▶ Sometida a constantes fuerzas de cambio.
- ▶ **Seis Grandes Ideas en Arquitectura de Computadoras:**
  - ▶ **Abstracción, Paralelismo, Pipelining, Predicción**, hacer **más rápido el caso más común** y usar una **Jerarquía de memorias**.