

Tema 01: Introducción a la Arquitectura de Computadoras

Arquitectura de Computadoras

Ing. Nicolás Majorel Padilla (npadilla@herrera.unt.edu.ar)

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

Temas que veremos

- ▶ **Conceptos básicos**
 - ▶ Partes de una computadora.
 - ▶ ¿Qué es Arquitectura de Computadoras?
 - ▶ Memorias: jerarquías, caché, Principio de localidad.
 - ▶ Modelos: de capas, abstracciones, fuerzas.
- ▶ **Relación con el Sistema Operativo**
 - ▶ Concepto de multitarea.
 - ▶ Modo Supervisor/Usuario.
 - ▶ Interrupciones.

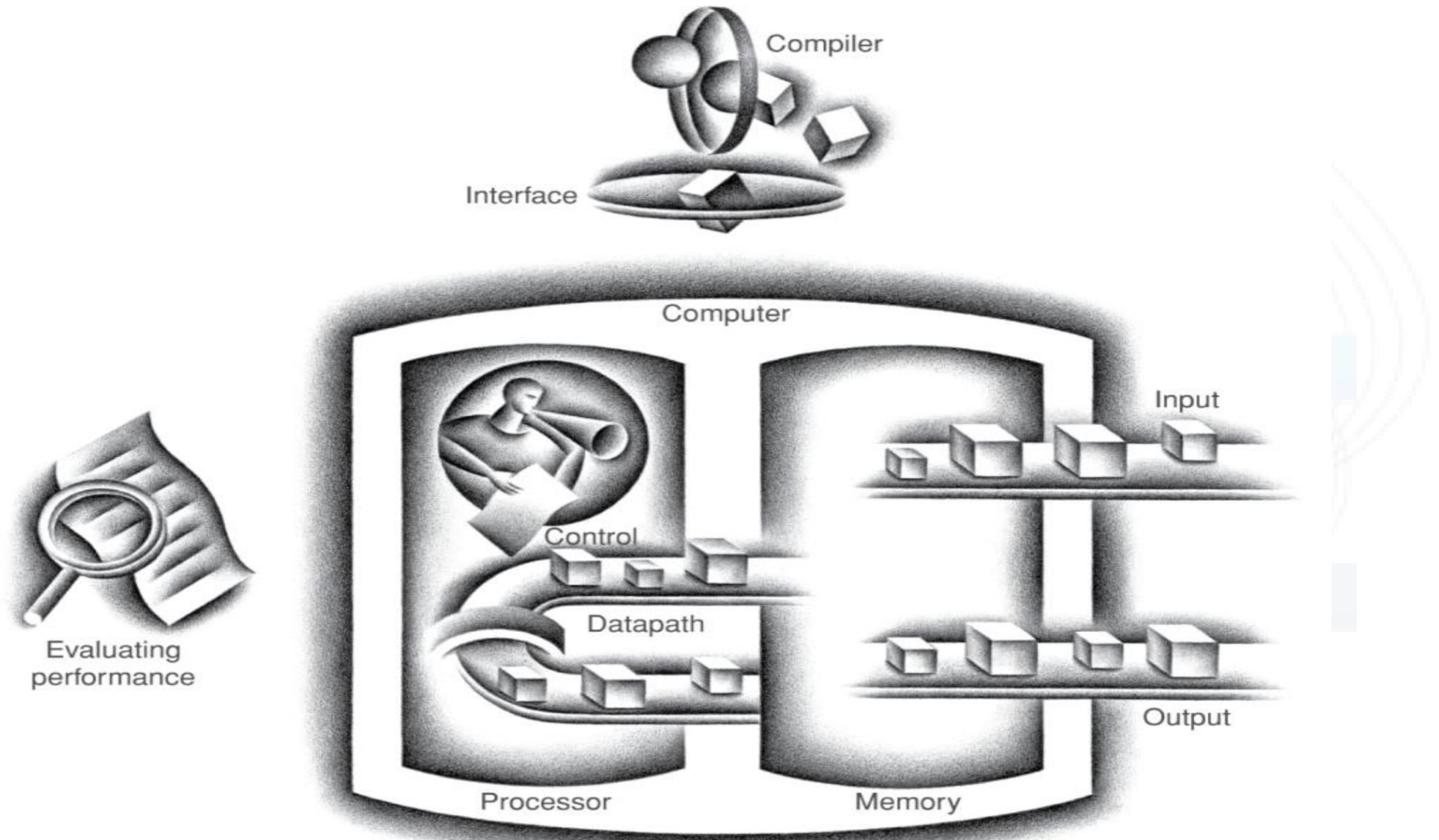
Lectura recomendada

- ▶ 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:
 - ▶ **PCs:** Propósito general. Software muy diverso.
 - ▶ Sujetas a compromiso entre costo y performance.
 - ▶ **Servidores:** pequeños, o como edificios.
 - ▶ Mayor capacidad, mayor performance, mayor confiabilidad.
 - ▶ **Supercomputadoras** (<http://top500.org>)
 - ▶ Grandes cálculos científicos.
 - ▶ Representan una fracción pequeña del mercado
 - ▶ **Embebidos**
 - ▶ Sujetos a estrictos compromisos entre costo, performance y consumo de energía.
- ▶ Nuevos tipos están empezando a surgir...

Partes de una computadora



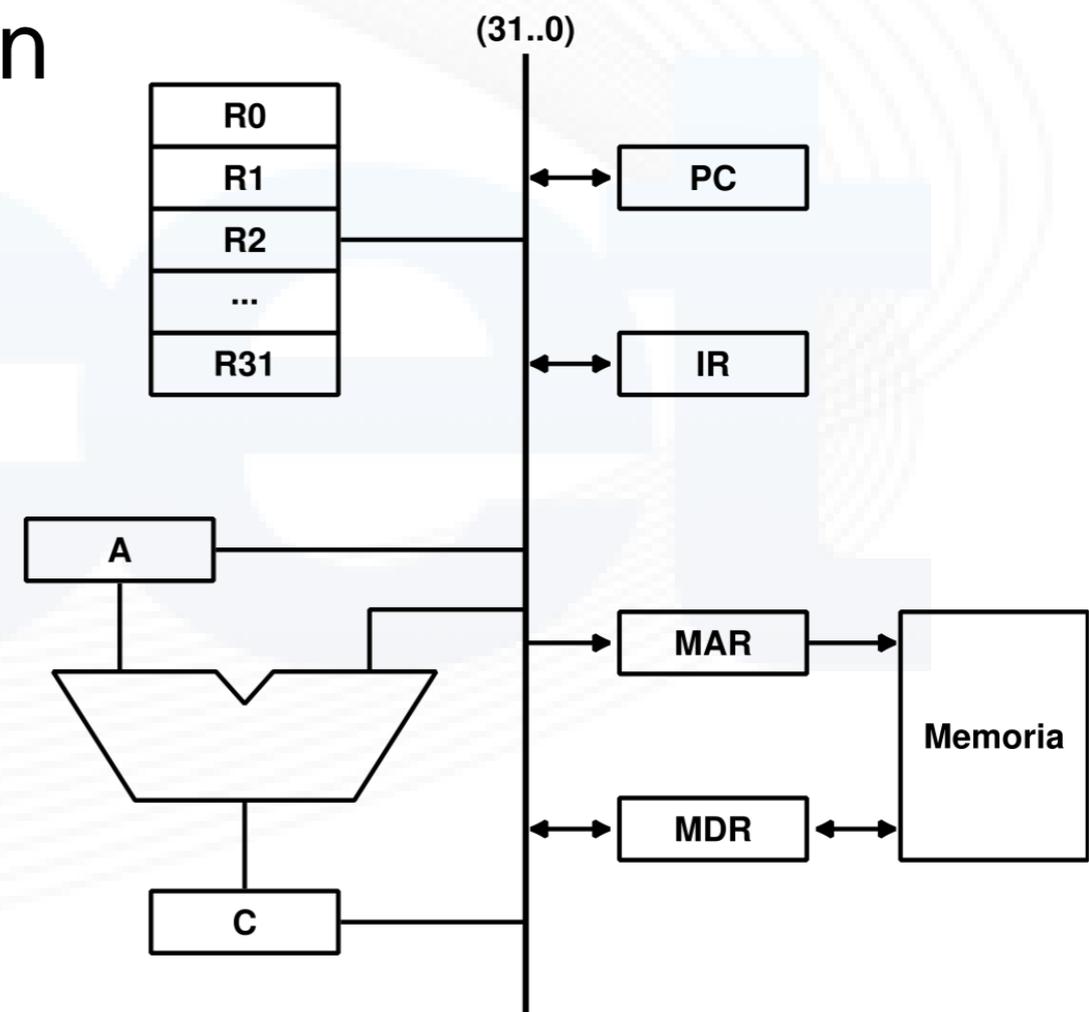
Partes de una computadora

- ▶ 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**.

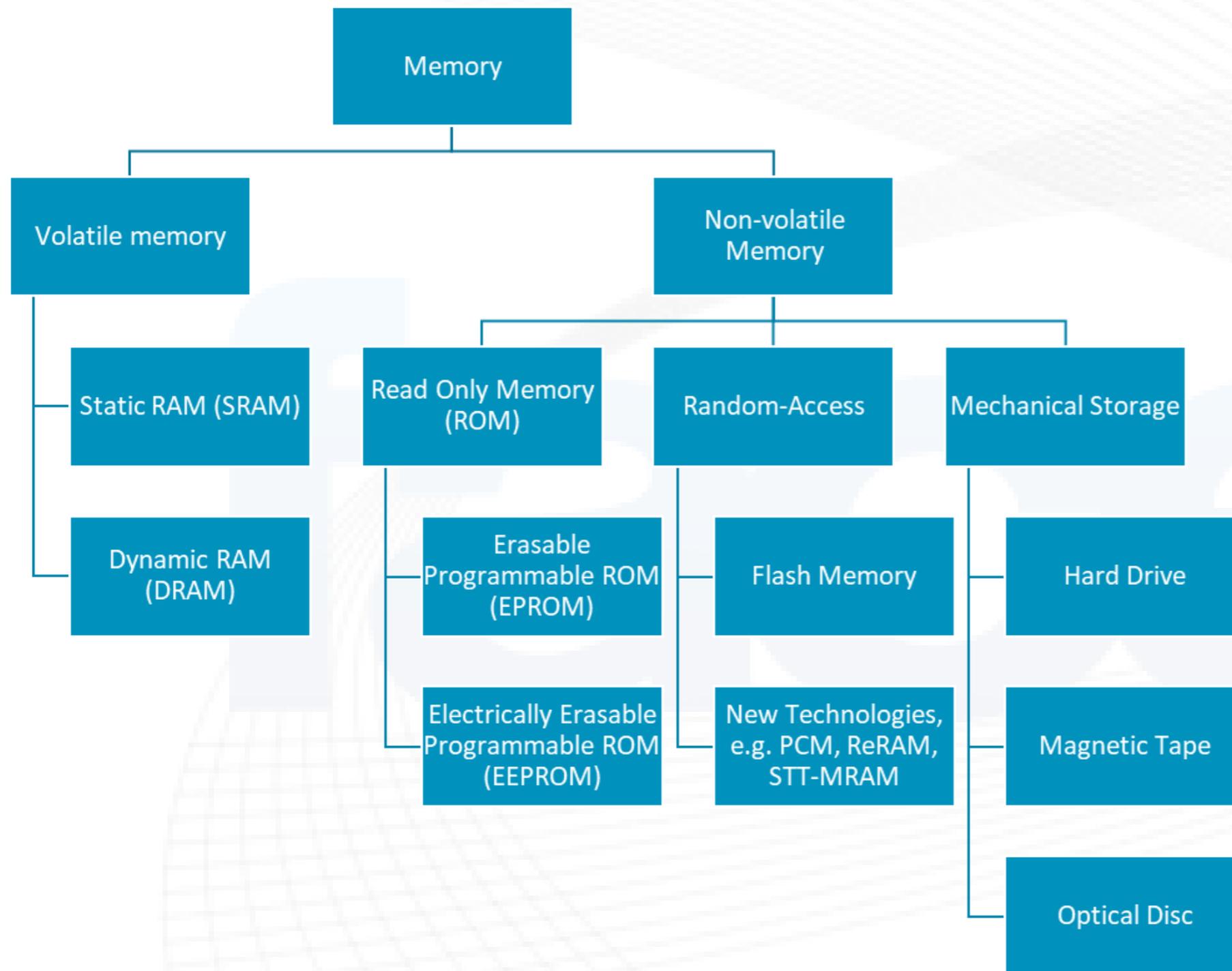
Procesador

- ▶ Ejecuta los programas.
- ▶ Ciclo de la instrucción
 - ▶ Fetch → Decode → Execution
- ▶ Ejemplo de SMM:

▶ **Repasar Tema 01 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...

Principio de Localidad – Definición

- ▶ Un programa accede a una porción relativamente pequeña del espacio de direcciones, en cualquier intervalo de tiempo.
- ▶ Dos tipos diferentes de localidad.
 - ▶ **Temporal:** cuando un ítem es accedido, es probable que pronto sea accedido nuevamente.
 - ▶ **Espacial:** cuando un ítem es accedido, es probable que pronto sean accedidos ítems cercanos.
- ▶ ¡Aplicable en casi cualquier ámbito!

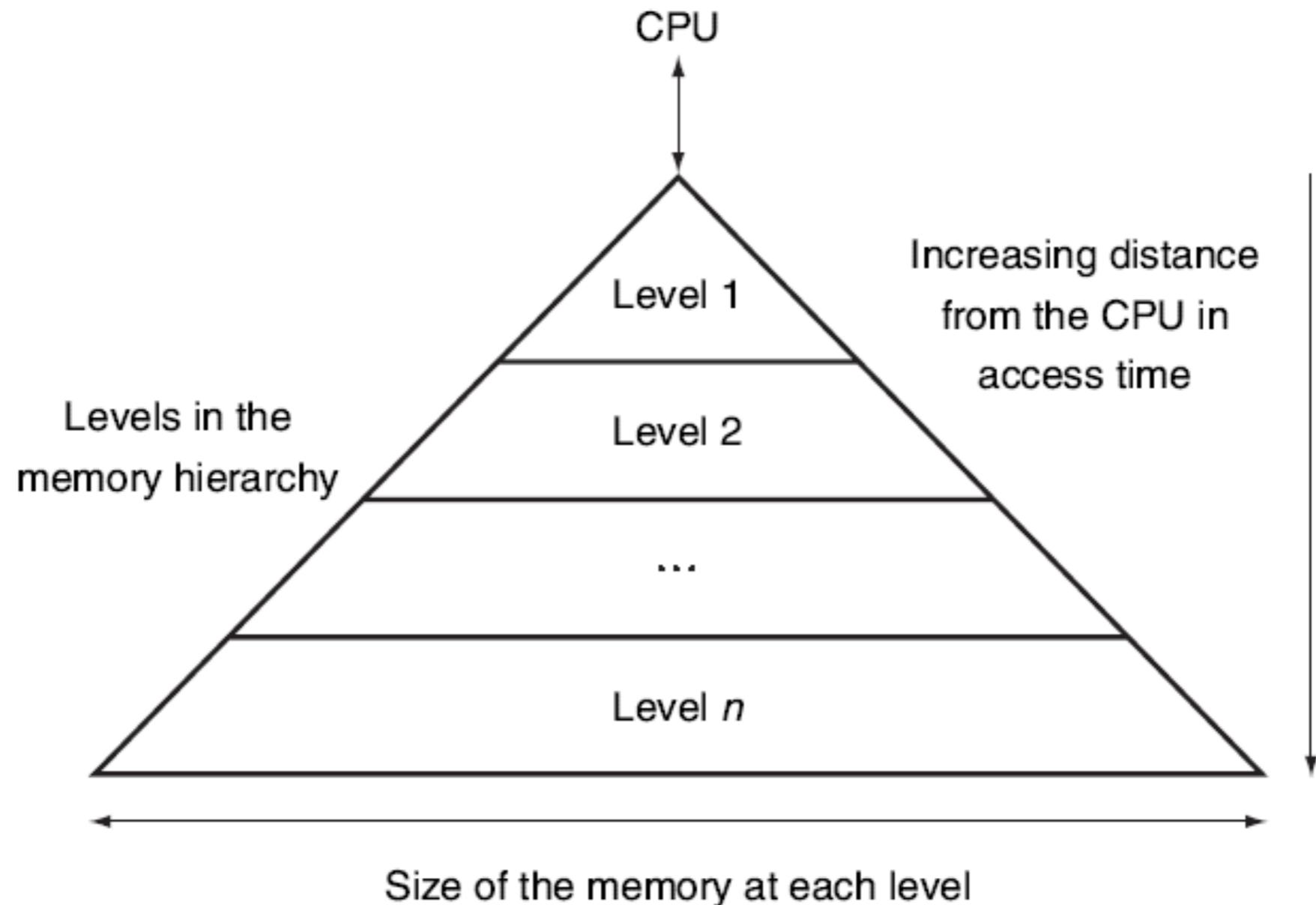
Principio de Localidad – Idea clave

- ▶ El principio de localidad nos da una idea clave:
 - ▶ La información que manejamos en un determinado intervalo de tiempo es acotada
 - ▶ ¡Ideal para memorias rápidas!
- ▶ ¡Las personas funcionamos así!
 - ▶ Cerebro → Notas rápidas → Apuntes → Libros → Bibliotecas
 - ▶ No es transparente (*todavía*).
 - ▶ No hay un sistema automático que detecte nuestra necesidad de información y la incorpore a nuestro cerebro sin que nos demos cuenta.

Pero...

- ▶ Normalmente, las memorias más rápidas (SRAM) son muy caras.
 - ▶ Y por ser caras, son pequeñas.
- ▶ Las memorias más baratas, son lentas.
 - ▶ Y por ser baratas, son grandes.
- ▶ **Objetivo:** brindar al usuario el mayor almacenamiento posible, lo más rápido posible (en promedio), con el menor costo posible. *¿Cómo?*

Jerarquía de Memorias



- ▶ Mayoría de accesos son a las memorias más rápidas.

Memoria caché – Idea clave

- ▶ Un **caché** es una memoria rápida, ubicada entre el procesador y la memoria principal.
- ▶ Pero la misma idea es aplicada entre memoria principal y disco.
 - ▶ Se denomina Memoria Virtual.
- ▶ Y también dentro de diversos dispositivos.
- ▶ Siempre son transparentes al usuario.
 - ▶ Manejados por Hw y por SO.

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%

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 RAM	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 RAM es **160 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.

Visión general

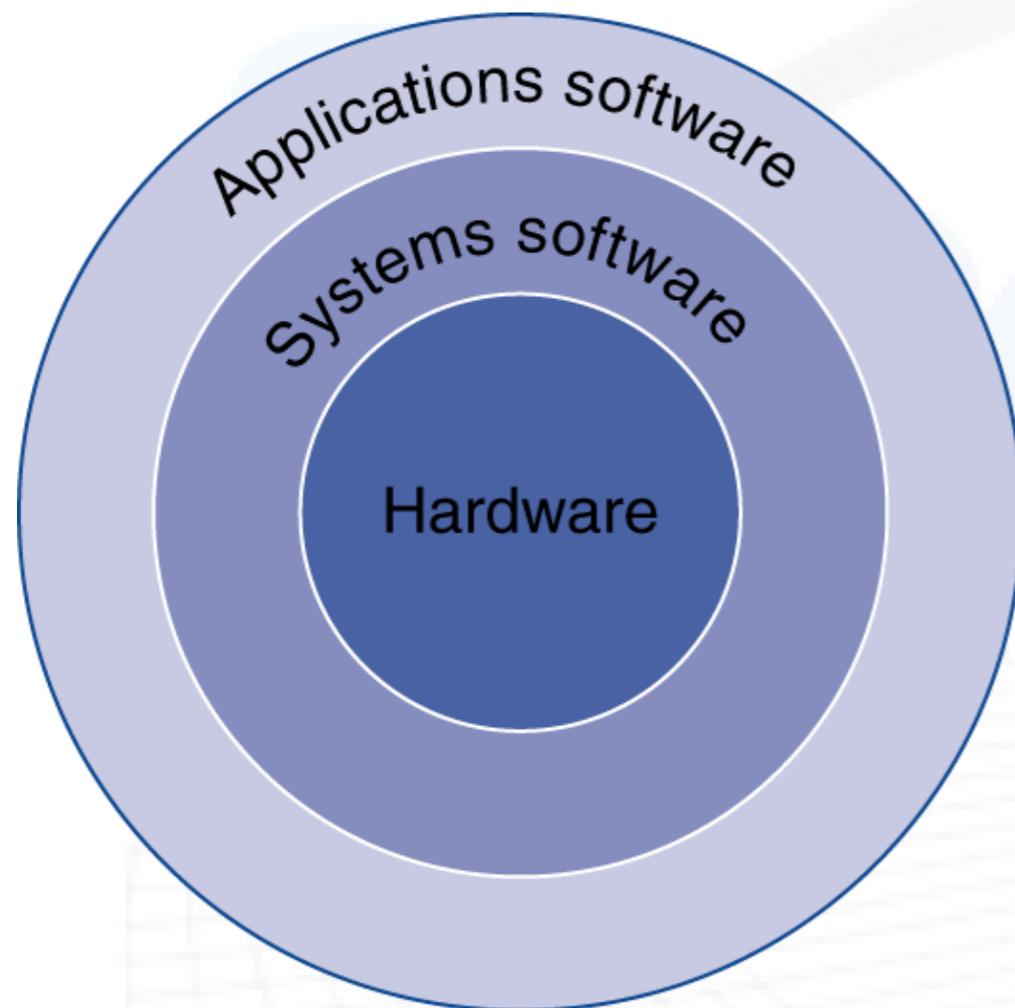
- ▶ Arquitectura de computadoras consiste fundamentalmente en **disimular las enormes diferencias de latencia/ancho de banda** de todas las partes que componen una computadora.
- ▶ Estas diferencias son los principales motivos de muchas técnicas que veremos durante el curso:
 - ▶ Paralelismo a nivel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

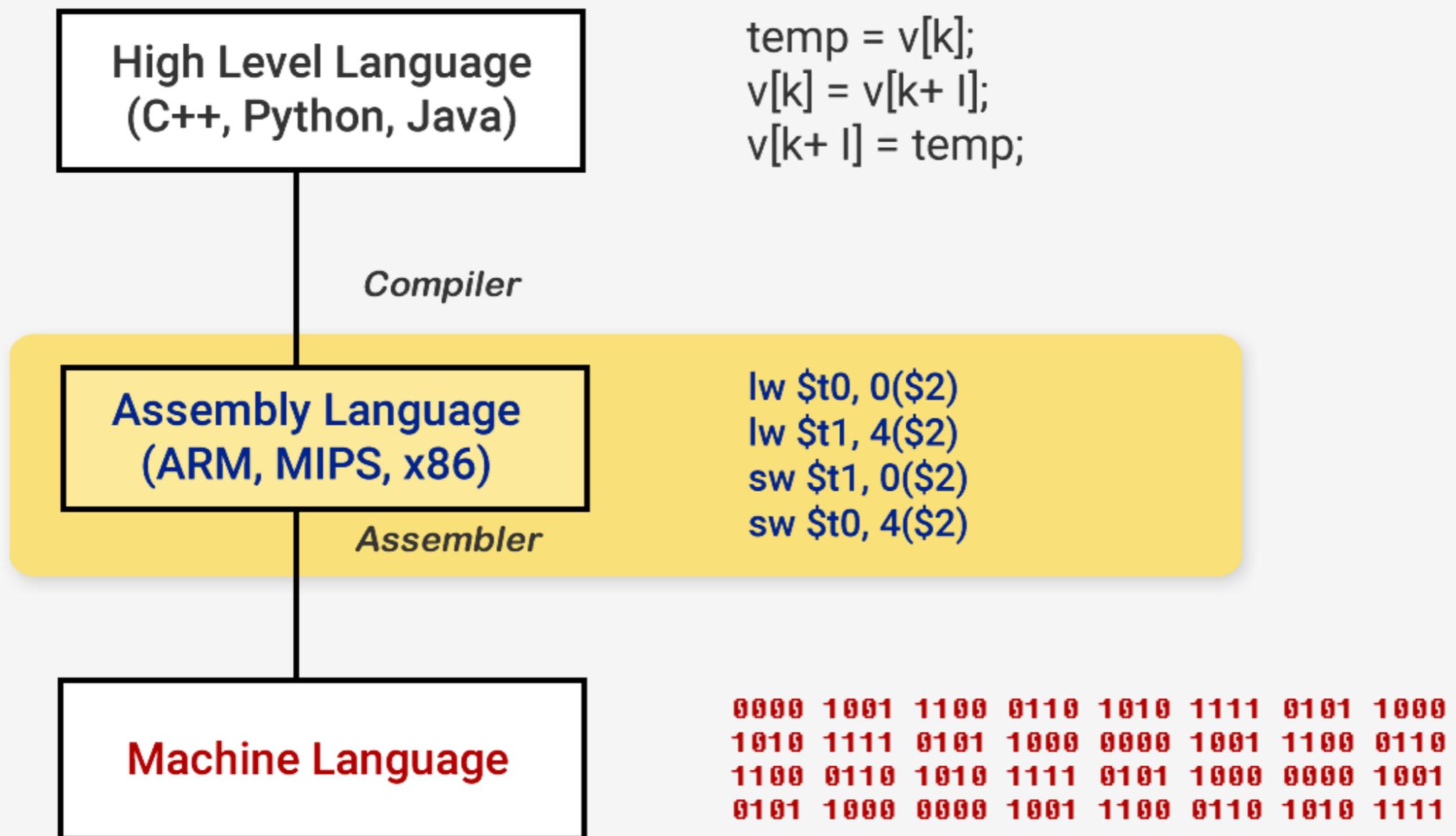
- ▶ **Permite manejar abstracciones.**
 - ▶ A través de **interfaces** entre ellas.



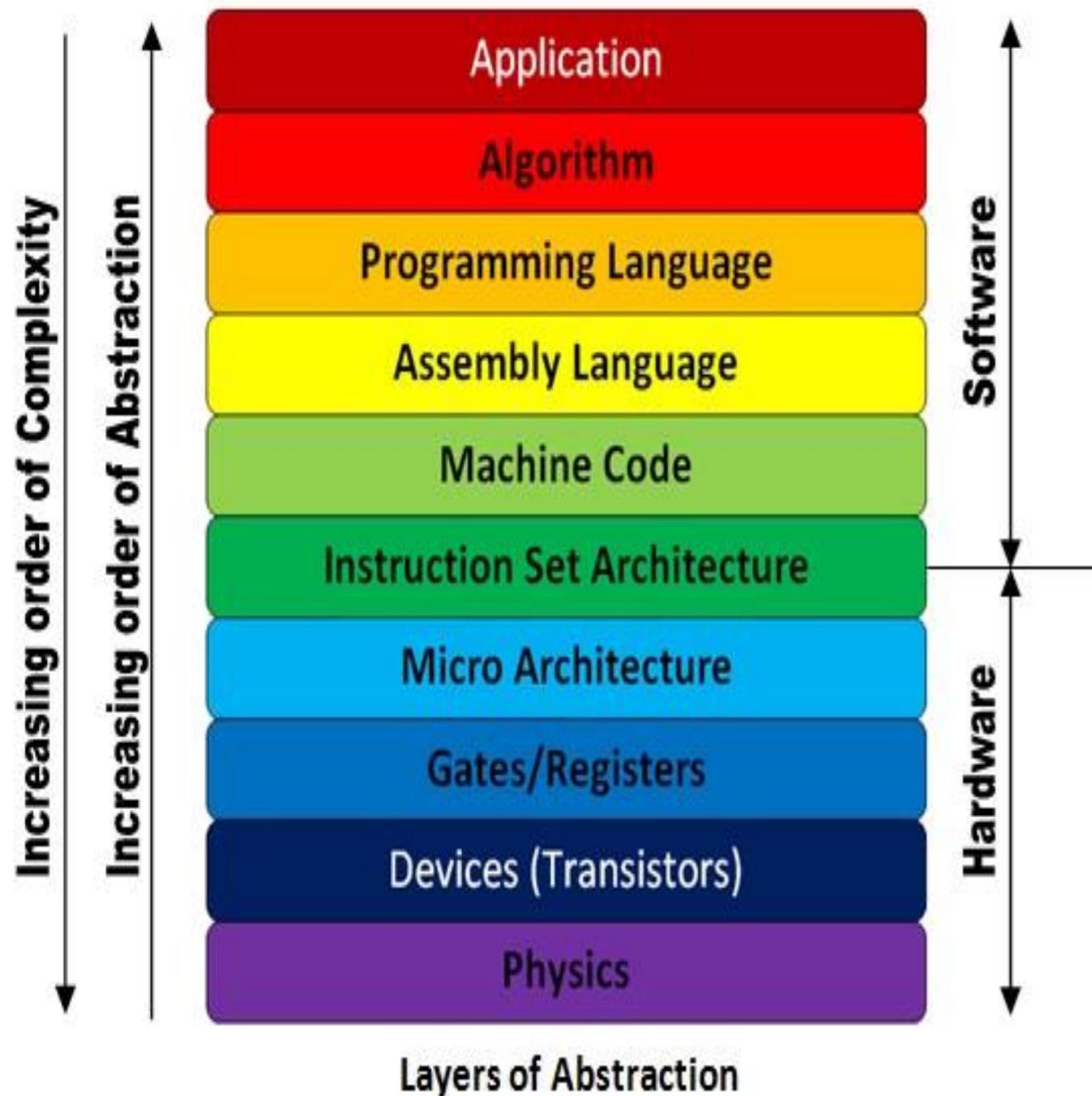
- ▶ **Software de sistema incluye:**
 - ▶ Sistema Operativo
 - ▶ Compilador
 - ▶ Enlazador
 - ▶ Linker/Loader
 - ▶ Ensamblador

Niveles de código

Levels of Program Code



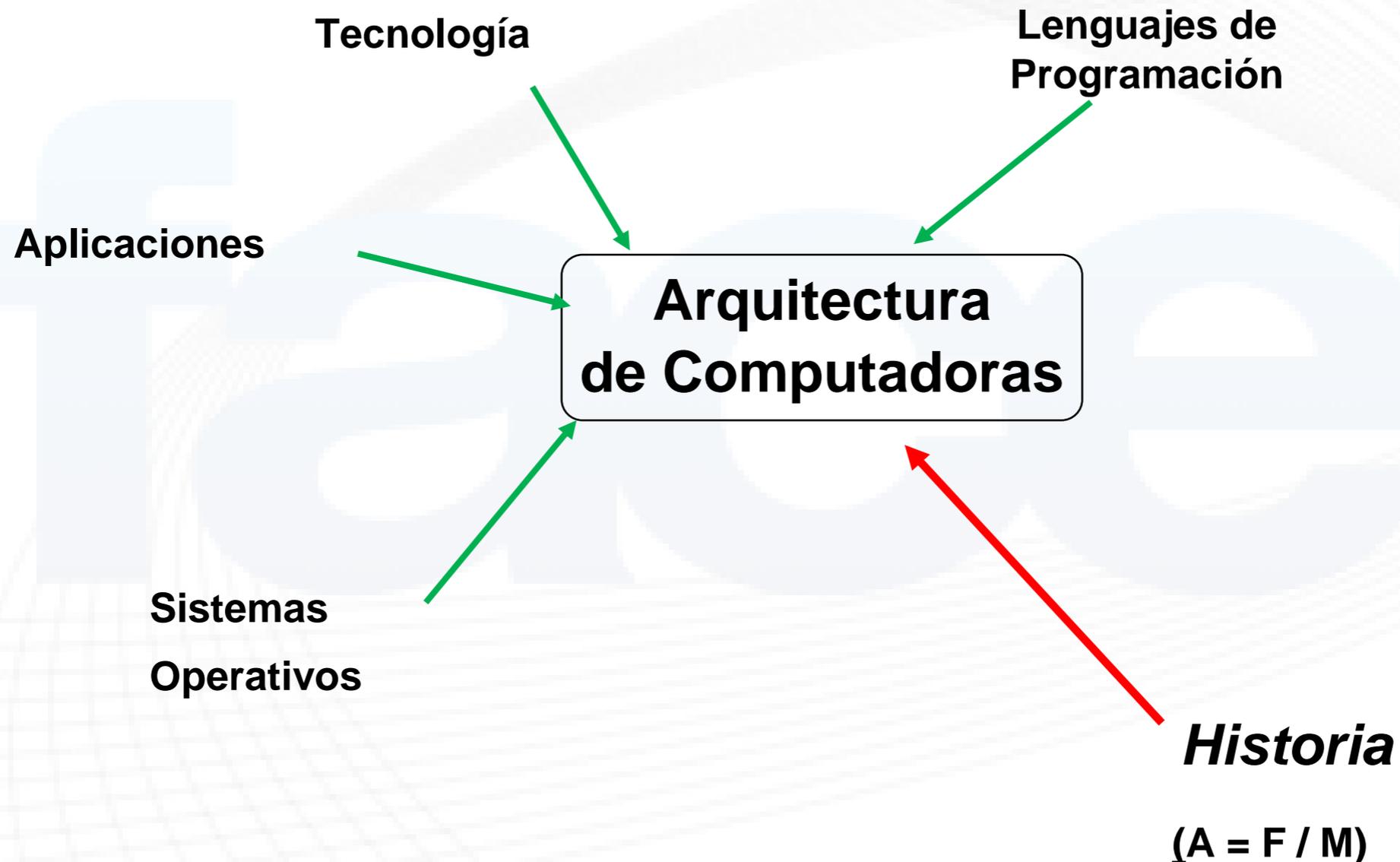
Modelo Jerárquico



- ▶ 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.

- ▶ ¡Fuerzas que cambian permanentemente!



Arquitectura de Computadoras en 2025

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

Sistema Operativo

- ▶ ¿Cómo se carga un programa en memoria?
¿Cómo se accede a un disco? ¿O al monitor?
 - ▶ No alcanza sólo con hardware.
- ▶ Se necesitan interfaces
 - ▶ Físicas y Lógicas.
 - ▶ Que sean transparentes al usuario.
- ▶ Además hay que optimizar el uso de los recursos, que son compartidos.
 - ▶ Que sea equitativo, seguro, confiable, transparente y simple de usar.
 - ▶ Recursos: procesador, memoria, disco, impresora, etc.

Multitarea – Concepto

- ▶ Un acceso a disco demora millones de ciclos del procesador.
 - ▶ Si se ejecuta un único programa, el procesador no hace nada durante el acceso al disco. **¡Muy ineficiente!**
 - ▶ Se podría hacer que otro programa use el procesador “al mismo tiempo”.
 - ▶ Para la eficiencia de todos los recursos, conviene que haya varios programas ejecutándose **concurrentemente**.
- ▶ Cada programa percibe que está solo.
 - ▶ Ningún programa puede afectar a los demás (**protecciones**).
- ▶ Individualmente, los programas pueden demorar más en terminar (**latencia**).
- ▶ Pero en promedio, muchos programas demoran menos (**productividad**).

Multitarea – Implementación

- ▶ Implica manejar:
 - ▶ Prioridades
 - ▶ Interrupciones
 - ▶ Modos de funcionamiento
- ▶ Todas estas son características del SO, pero que **deben ser soportadas por el Hw debajo.**
 - ▶ *Uds. ya vieron algo en SMM, ¿no?*
- ▶ Las excepciones le permiten al SO tomar acciones en respuesta a eventos o pedidos que ocurren cuando un programa de usuario está ejecutándose.
 - ▶ Comienza desde el manejador de interrupciones (*handler*).
 - ▶ Los programas de usuario pueden solicitar servicios al SO.

Modos de funcionamiento

- ▶ En general se manejan 2 modos:
 - ▶ **Usuario** (con restricciones).
 - ▶ **Kernel/Supervisor** (sin restricciones).
 - ▶ *¿En qué modo trabaja el SO?*
- ▶ Arquitecturas más nuevas manejan varios niveles más, pero son variantes de estos dos.
 - ▶ Por ejemplo, el procesador que veremos en el curso maneja tres.
- ▶ El estado del procesador se mantiene en un registro especial.
 - ▶ Contiene bits que indican el modo que se está ejecutando.
- ▶ Hay instrucciones del procesador “privilegiadas”
 - ▶ Sólo se ejecutan en modo kernel.

Resumen final

- ▶ Hay cuatro tipos principales de computadoras
- ▶ **Pero todas están compuestas por los mismos 5 componentes.**
 - ▶ Procesador (Camino de Datos y Control), Memoria, Entrada y Salida.
- ▶ Principio de Localidad
 - ▶ Temporal y Espacial
- ▶ Concepto de **Jerarquía de Memorias**
 - ▶ Presentar la ilusión de una memoria muy grande, barata, y muy rápida.
- ▶ Concepto de **multitarea**
 - ▶ No perder tiempo esperando.
 - ▶ Sacrificar un poco de **latencia** para ganar **productividad**.

Resumen final

- ▶ **Rol del Sistema Operativo**
 - ▶ Manejar de manera eficiente los recursos compartidos.
 - ▶ Presentar interfaces más simples a los usuarios.
 - ▶ Proveer mecanismos de seguridad y protección.
- ▶ **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.

Agradecimientos

- ▶ Las diapositivas de este tema fueron basadas en las realizadas por el Ing. Daniel Cohen.
- ▶ A su vez inspiradas en las clases del curso CS152 de la Universidad de Berkeley, California, USA.
 - ▶ Realizadas por los Prof. D. A. Patterson, John Lazzaro, Krste Asanovic.