Unidad 1 — Introducción

Escuela Técnica Nº 35 "Ing. E. Latzina"
Santiago Trini

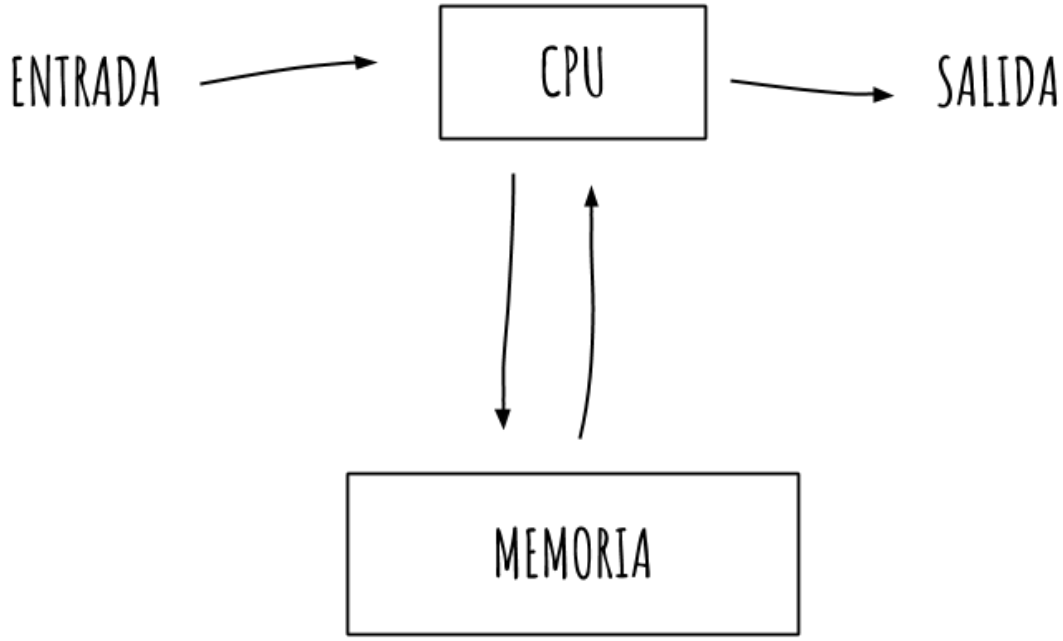
Organización y arquitectura de computadoras

Esta materia aborda la relación entre *hardware* y *software* e intenta contestar algunas preguntas referidas al diseño concreto de computadoras físicas en el mundo real. Hablamos de *software* cuando nos referimos a programas, sistemas operativos, aplicaciones y cualquier parte en una computadora que no podemos tocar. Es lo que generalmente escriben los programadores. En contraste, el *hardware* son los circuitos electrónicos y dispositivos mecánicos que interpretan las instrucciones del software. Son las cosas que podemos ver y tocar dentro de una computadora. La distinción entre *software* y *hardware* no siempre es tan clara, porque el *hardware* también implementa instrucciones.

La **organización de computadoras** estudia las partes funcionales de la computadora y como interactúan entre sí. Ayuda a contestar la pregunta: ¿cómo funciona una computadora? La **arquitectura de computadoras** estudia las instrucciones que ejecuta el *hardware*, los mecanismos que implementan los circuitos electrónicos para realizar más operaciones y más rápido. Ayuda a contestar la pregunta: ¿cómo diseño una computadora? Estas dos disciplinas están estrechamente relacionadas y generalmente se estudian juntas.

Definición de computadora

Una computadora está, en esencia, compuesta por tres partes. Una **CPU** (unidad central de procesamiento), una **memoria**, y un **sistema de E/S** (entrada y salida). Esa es la definición más general que podemos dar, pero también entendemos generalmente que una computadora es una máquina de **propósito general**, que es **programable**, y que es un dispositivo **electrónico**. No todas las computadoras tienen que ser electrónicas necesariamente, de hecho existen computadoras mecánicas y electromecánicas, pero son mucho más lentas. Por último, seguramente cuando hablamos de computadoras estamos hablando de circuitos **digitales**. Esto tampoco es necesario, pero es cierto en la gran mayoría de las computadoras. Cuando hablamos de computadoras digitales nos referimos a que una computadora opera con valores discretos, no continuos. De ahí que decimos que una computadora trabaja con unos y ceros (código binario).



Las tres partes funcionales de una computadora.

Niveles de abstracción

Una forma de reducir la complejidad de un sistema o computadora es ir agregando **capas de abstracción**. Esto en informática es bastante común, no sólo cuando hablamos de diseñar computadoras. En organización de computadoras hablamos de las siguientes capas o niveles de abstracción.

Nivel 0 — Lógica digital

Este nivel se implementa directamente con circuitos electrónicos. Es donde encontramos el *hardware*, lo que podemos tocar realmente. En definitiva en este nivel tenemos **compuertas** y cables. Las compuertas se implementan hoy en día con **transistores**. El funcionamiento de los dispositivos eléctricos y electrónicos que forman parte de este nivel corresponde a la especialidad de la electrónica y no de la computación.

Nivel 1 — Control o microarquitectura

El nivel de control o microarquitectura se encarga de llevar a cabo las instrucciones del nivel 2 en el *hardware*. La implementación de este nivel puede ser directamente con compuertas, en este caso se dice que el control es *hardwired*. O bien puede ser realizada con una técnica conocida como microprogramación.

Nivel 2 — Lenguaje máquina o arquitectura

El lenguaje máquina es el conjunto de instrucciones que puede interpretar una computadora. Es específico de cada arquitectura, como cuándo hablamos de x86, ARM, AVR, etc. Si el nivel de control está implementado directamente sobre la electrónica, sin mediación de un microprograma, entonces las instrucciones de lenguaje máquina pueden ser interpretadas directamente por los circuitos electrónicos.

Nivel 3 — Software de sistema

Este nivel es donde se ubica el sistema operativo. Es un software con acceso privilegiado al *hardware* y cualquier otro programa que se ejecute en una computadora accede al *hardware* a través de él.

Nivel 4 — Lenguaje ensamblador

Un lenguaje ensamblador o *assembler* es un lenguaje que tiene una correspondencia uno a uno con las instrucciones del lenguaje máquina. Los compiladores que usamos con los lenguajes de alto nivel producen código ensamblador para la arquitectura con la que estamos trabajando. Programar en *assembler* es difícil comparado con un lenguaje de alto nivel, pero mucho menos propenso a errores que usando el lenguaje máquina directamente. En el lenguaje máquina tenemos que lidiar con números y en cambio en ensamblador reemplazamos los códigos numéricos de las instrucciones con palabras como `ADD`, `MOV`, etc.

Nivel 5 — Lenguajes de alto nivel

Estos lenguajes de programación están orientados a un dominio específico de problemas y están mucho más cerca del lenguaje natural que del lenguaje máquina. Los programas en lenguajes de alto nivel deben ser traducidos (compilados o interpretados) para que el *hardware* pueda ejecutarlos. Algunos ejemplos de lenguajes de alto nivel son: Java, C++, Python, Javascript y Ruby.

Arquitectura de von Neumann

Un concepto importante para entender el funcionamiento de una computadora es el concepto de **computadora de programa almacenado**. En las primeras computadoras como la ENIAC programar era sinónimo de conectar cables y clavijas al estilo de las viejas operadoras telefónicas. Reprogramar una máquina era recablearla para que realice una operación distinta.

En los años que se preparaba el sucesor de la ENIAC, la EDVAC, comenzó a surgir la idea de usar la memoria no sólo para almacenar los datos sobre los que un programa opera, sino también las instrucciones del programa. Un *paper* del famoso matemático John von Neumann popularizó la idea y por eso se conoce hoy como arquitectura de von Neumann.

La idea es que una computadora son los tres sistemas que ya mencionamos: una CPU, una memoria principal y un sistema de entrada y salida. A su vez la CPU consiste de tres partes: una **ALU** (*Arithmetic Logic Unit*), una **CU** (*Control Unit*) y **registros** (espacios de memoria dentro de la CPU). Uno de estos registros es muy importante y recibe el nombre especial de *Program Counter* (contador de programa).

Contador de programa

El contador de programa es un registro que guarda la dirección de memoria de la próxima instrucción a ejecutar.

Ciclo de instrucción

Desde que se apreta el botón de encendido hasta que se apaga la computadora, la CPU ejecuta una y otra vez la misma operación de manera repetitiva. Esta operación se conoce como ciclo de instrucción y es la base para entender como funciona una computadora. El ciclo de instrucción se

conoce también como ciclo *fetch-decode-execute* o simplemente ciclo *fetch-execute*. Estos nombres indican las fases del ciclo de instrucción. Hay dos unidades funcionales privilegiadas en una computadora, la CPU y la memoria principal. La interacción entre las dos se da de la siguiente manera:

1. La CPU carga en un registro especial el o los bytes a los que hace referencia el PC. Este registro toma varios nombres, pero lo podemos llamar IR (*Instruction Register*). Lo que se carga en IR es una cadena de bits que está en la memoria y cuya dirección está apuntada por el contador de programa. Esta fase se conoce como *fetch*
2. La CPU decodifica la cadena de bits que ahora está en IR interpretándola como una instrucción. Esta fase es *decode*.
3. La CPU ejecuta la instrucción activando distintas señales de control según el propósito de la instrucción. Por ejemplo, puede activar señales de control en la ALU para realizar una operación aritmética, o activar señales de control en la memoria principal para guardar una palabra. Esta es la fase de *execute*.

Esto se vuelve a repetir una y otra vez hasta que se apague la computadora.

Unidades de medida

En computación usamos los prefijos del Sistema Internacional de Unidades que se resumen en la siguiente tabla.

Prefijo	**Símbolo**	**10^n**
exa	E	10^{18}
peta	P	10^{15}
tera	T	10^{12}
giga	G	10^{9}
mega	M	10^{6}
kilo	K	10^{3}

Las potencias negativas de 10 rara vez se usan (mili, micro) porque o no tienen sentido o no se encuentran en la práctica. Cuando hablamos de información hablamos de **bytes** y no tiene ningún sentido dar fracciones de bytes. Cuando hablamos de velocidad nos referimos en ciclos por segundo o **hertz** y en la práctica siempre tenemos velocidades en el rango de los megahertz.

Sin embargo al hablar de almacenamiento de información muchas veces se produce ambiguedad debido a que tiene mucho más sentido contar en base 2 que en base 10. Por ejemplo, 1 KB (kilobyte) ¿son 1000 bytes o 1024 bytes?

En realidad por como se construyen las memorias generalmente el número de bytes es una potencia de dos. Para evitar confusiones, cuando hablamos de almacenamiento conviene utilizar los prefijos que indican explícitamente que estamos usando base dos. Al kilobyte = 1 KB = 1000 bytes = 10^3 bytes le corresponde el kibibyte = 1 KiB = 1024 bytes = 2^{10} bytes. No hay una equivalencia estricta entre las unidades en base 10 y base 2, pero sí aproximada ya que $2^{10} = 1024 \approx 10^3$,

$2^{20} = 1048576 \approx 10^6$, etc. Se resumen en la siguiente tabla.

Prefijo	Símbolo	2^n
exbi	Ei	2^{60}
pebi	Pi	2^{50}
tebi	Ti	2^{40}
gibi	Gi	2^{30}
mebi	Mi	2^{20}
kibi	Ki	2^{10}

Variables de tiempo y espacio en una computadora

Cuando hablamos de tiempo y espacio en computación nos referimos a la velocidad a la que opera una computadora y la cantidad de información que puede almacenar. La velocidad de una computadora es en esencia la cantidad de instrucciones que puede ejecutar por unidad de tiempo, aunque no hay una sola forma de medir esto. La cantidad de almacenamiento se mide en bits y sus múltiplos, porque en esencia todo la información almacenada son números en binario.

Bits y bytes

Un poco de terminología: un bit es la unidad mínima de información, un uno o un cero. Un byte por razones históricas decantó en la suma de 8 bits. Cuando hablamos de memoria en general por el método de fabricación electrónica tiene más sentido usar los múltiplos de base binaria. Por lo tanto, aunque a veces se produce confusión, es conveniente decir 1 KiB para referirse a 1024 bytes y no un KB que estrictamente serían 1000 bytes. 1 MiB son 1024 KiB, 1 GiB son 1024 MiB y así sucesivamente. Muchas veces se usa KB y KiB o los otros múltiplos indistintamente.

Tiempo

La primer medida de la velocidad que suele presentarse es la frecuencia de reloj, generalmente de la CPU en Hz o sus múltiplos. Por ejemplo decimos que una computadora es más rápida que otra porque una tiene un microprocesador 2 GHz más veloz que la otra. Esto es acertado sólo en parte, ya que hay muchas más variables que determinan la velocidad de ejecución de un programa.

La primera y más obvia es el programa mismo que se ejecuta. Muchas veces entendemos a un programa como la tarea que realiza y no las instrucciones que ejecuta. Esto implica que un algoritmo más eficiente y una programación mejor hecha por supuesto mejora el tiempo de ejecución de una tarea.

Otra menos obvia es el diseño mismo de la CPU, su microarquitectura. Supongamos dos microprocesadores corriendo a la misma frecuencia y con el mismo conjunto de instrucciones. Pero el diseño interno de un procesador le permite ejecutar dos instrucciones por ciclo de reloj contra una instrucción por el otro. Por lo tanto una medida que a veces se toma como sinónimo de desempeño son los MIPS (millones de instrucciones por segundo). Los MIPS están relacionados con la frecuencia y con la cantidad de instrucciones que ejecuta una CPU por ciclo de reloj llamado generalmente IPC (instrucciones por ciclo). Las IPC son el inverso multiplicativo de las CPI (ciclos

por instrucción) que es una medida de desempeño más utilizada en la literatura técnica.

Los MIPS y CPI son buenas medidas de desempeño pero más bien anticuadas, están ligadas al desempeño de una CPU con números enteros. Actualmente las operaciones de coma flotante tienen más importancia y se suele dar un número que indica la velocidad con la que la CPU procesa este tipo de operaciones. Esta medida se conoce como FLOPS (*floating point operations per second*) y se expresa utilizando múltiplos como MFLOS (megaflops), GFLOPS y TFLOPS. Las GPUs (*graphic processing units*) modernas ya exceden la barrera del TFLOP (10^{12} operaciones de coma flotante por segundo).

Evolución histórica de la computadora

En esta sección hacemos un breve repaso por la evolución histórica de la computadora. La importancia de conocer brevemente la historia de la computadora es que nos ayuda a comprender mejor el funcionamiento de la misma y las abstracciones más comunes que perduran al día de hoy, más allá de los cambios en la tecnología que se producen a un ritmo bastante acelerado.

La mayoría de los manuales coinciden en agrupar a las distintas computadoras en **generaciones** basadas en la tecnología disponible en el momento para implementar los circuitos digitales básicos que constituyen las distintas partes de una computadora programable.

Generación cero (1642 - 1945)

Estas primeras máquinas de calcular y computadoras no eran electrónicas, sino mecánicas y luego electromecánicas. No eran computadoras programables en el sentido que entendemos hoy en día, excepto por contadas excepciones como la Z1 de Konrad Zuse o la Máquina Analítica de Charles Babbage. Es con esta última máquina que se reconoce la invención del primer programa de computadora y a la primer programadora: Ada Lovelace.

Generación uno (1945 - 1953)

La primer generación de computadoras está caracterizada por utilizar **tubos de vacío**. Un tubo de vacío puede ser utilizado como un interruptor controlado electrónicamente y es el antecesor directo del transistor.

La ENIAC es considerada la primer computadora en el sentido moderno del término. A partir de esta generación todas las computadoras que le siguen son electrónicas y de propósito general. El paso a las computadoras electrónicas permitió una velocidad de operación sin precedentes.

Por ejemplo, la ENIAC se creó originalmente para calcular tablas balísticas para el ejército y el cálculo de una tabla que antes llevaba 20 horas se acortó a 30 segundos.

Generación dos (1954 - 1965)

En 1948 se inventó el **transistor** y poco tiempo después esta nueva tecnología reemplazó a los tubos de vacío en las computadoras. Esto permitió construir computadoras más pequeñas y que consumieran menos energía. La ENIAC por ejemplo consumía 174 kilowatts y ocupaba 168 metros cuadrados. Como los transistores consumían menos energía también se calentaban menos y permitían mayores velocidades. Además el cambio de tecnología bajó los precios.

Por ejemplo, la IBM 7090 era seis veces más rápida que su antecesora con tubos de vacío, y podía ser alquilada por la mitad de su precio.

Generación tres (1965 - 1980)

En 1959 Jack Kilby inventa el **circuito integrado**. Este nuevo método de fabricación consiste en incorporar los transistores y otros componentes electrónicos en un área pequeña de material semiconductor mediante **fotolitografía**. Por lo tanto los transistores se miniaturizan y consumen aún menos energía que en sus versiones discretas.

Esto se tradujo en menores costos, más velocidad y menor tamaño. La tendencia es clara. En esta generación las computadoras comenzaron a ser más accesibles y ya se podían encontrar no sólo en grandes empresas multinacionales o agencias de gobierno, sino también en universidades aunque no aún en los hogares. Algunos ejemplos de esta generación son la IBM System/360 o la PDP-11. Esta última fue la primer computadora en correr UNIX.

Generación cuatro (1980 - actualidad)

Los circuitos integrados se clasifican por su **nivel de integración**:

SSI

(*Small Scale Integration*) 10 a 100 transistores

MSI

(*Medium Scale Integration*) 100 a 1000 transistores

LSI

(*Large Scale Integration*) 1000 a 10000 transistores

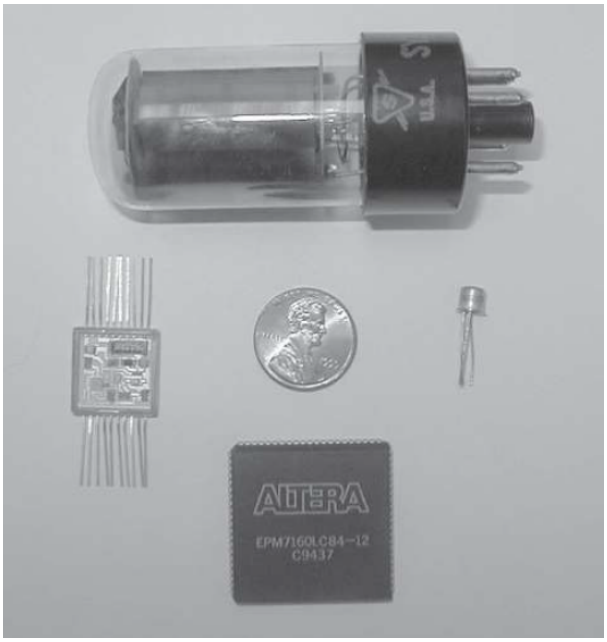
VLSI

(*Very Large Scale Integration*) 10000 a 100000 transistores

Esta generación no se caracteriza por un cambio de tecnología, aunque generalmente se asocia con los circuitos VLSI y superiores. Más importante es la aparición de la PC de IBM (computadora personal) en 1980. La difusión de los sistemas operativos, las interfaces gráficas y el mouse que aparecieron con la Apple Lisa.

La importancia de la PC de IBM fue que la compañía esta vez, ya en su tercer intento por producir una máquina para el público en general que fuera accesible liberó la arquitectura de la computadora a sus competidores. Esto permitió que la competencia fabrique clones de la PC y creó *de facto* un estándar que perdura al día de hoy. De la mano de la PC también se formaron dos monopolios importantes en la industria, el de Microsoft como proveedor de sistema operativo para la PC (primero con DOS y luego con Windows), y el de Intel como el líder indiscutido en la fabricación de microprocesadores para estas nuevas máquinas.

Para poner en perspectiva, en 1997 se realizó una copia de la ENIAC pero en un único circuito integrado. Las 30 toneladas de la ENIAC se redujeron a un sólo integrado del tamaño de una uña con 174569 transistores.



Comparación entre tecnologías, en sentido horario: tubo de vacío, transistor, integrado con 3200 compuertas NAND, varios circuitos integrados en un solo empaquetado sin la cubierta cerámica.

Ley de Moore

La ley de Moore es una observación empírica formulada por Gordon Moore, uno de los fundadores de Intel. Lo que la experiencia demuestra es que la cantidad de transistores en un circuito integrado se duplica cada 18 meses.

Esto por supuesto no se va a mantener de manera indefinida, no se pueden fabricar transistores más pequeños que un átomo, pero cuando Moore hizo su afirmación original pensó que se cumpliría por diez años. Casi 55 años después la ley de Moore sigue en pie, aunque ya mostrando signos de desaceleramiento.

Partes de una PC

La PC o computadora personal nació en los años '80 y sigue siendo un estándar en uso hoy en día. A continuación repasamos brevemente los elementos de *hardware* típicos en una PC de escritorio.

Placa base

El *motherboard* o placa base cumple con diversas funciones dentro de una computadora. En esencia sirve como un circuito que conecta las tres partes de una computadora: memoria, CPU y E/S. Pero en su evolución histórica fue adquiriendo más o menos características propias, generalmente en relación con el sistema de E/S, y en menor medida con la comunicación directa entre CPU y memoria.

Microprocesador

En todas las PC un microprocesador es el circuito integrado que cumple la función de CPU. De todos los circuitos integrados que integran una computadora el microprocesador es lejos el más complejo y con más cantidad de transistores por área. Para entender la diferencia entre un microprocesador y una CPU, esta última es parte de cualquier computadora, pero en la década del '70 no era posible

incluir todos los elementos que hacen una CPU en un solo circuito integrado. En 1971 Intel consigue fabricar el 4004, el primer microprocesador desarrollado para una calculadora, y en 1974 el 8080, el primer microprocesador de uso general pensado como la CPU de una computadora. El 8080 tenía 4500 transistores en un solo circuito integrado.

Ejemplos de microprocesadores modernos son el AMD Ryzen, el Intel Core i7, el SPARC M8 (arquitectura SPARC para servidores) o el Broadcom BCM2837 (arquitectura ARM, el del Raspberry Pi 3).

Al ser la implementación de la CPU, el microprocesador determina la mayoría de las características de una computadora como su lenguaje máquina o arquitectura, sus buses, el espacio de memoria direccionable, etc.

Una característica importante de un microprocesador es la **frecuencia** a la que opera, es decir la cantidad de ciclos por segundo de su reloj. Esto determina en parte la cantidad de instrucciones que ejecuta por segundo. A más frecuencia también se genera más calor y se consume más energía, por este motivo los procesadores modernos pueden ajustar su frecuencia de manera variable. Los microprocesadores de las PC modernas operan en el rango de los GHz.

Es indispensable que un microprocesador esté acompañado de un *cooler* para disipar el calor que genera, y hoy en día hay dos soluciones en las PC de escritorio. Disipadores por aire con un disipador metálico de cobre o estaño y un ventilador, y refrigeración líquida.

Memoria RAM

La memoria RAM de *random access memory* es el componente de *hardware* que implementa la memoria principal de una computadora. Se denomina de acceso aleatorio porque acceder a una posición en la memoria tiene el mismo tiempo de espera para cualquier dirección, en contraste por ejemplo con un disco rígido.

Hay dos tipos principales de tecnología para fabricar los integrados de memoria: SRAM y DRAM.

La SRAM *static RAM* es una tecnología basada completamente en transistores pero no se usa en las tarjetas de memoria que encontramos en las PC. La SRAM tiene un precio elevado por bit y se reserva para implementar la memoria cache del procesador. Además la SRAM no tiene por qué ser volátil, es decir puede no borrarse cuando se apaga la computadora y existen las dos variantes. Cada celda de memoria SRAM (1 bit) usa aproximadamente seis transistores.

En cambio la DRAM *dynamic RAM* usa un transistor y un capacitor por bit. Es mucho más económica por bit que la SRAM pero no puede ser no volátil. Al desconectar la energía sus celdas se blanquean. Se denomina dinámica por que necesita **refrescarse** periódicamente. Esto es porque cada bit se almacena en un capacitor que se descarga de a poco, por lo tanto la memoria dinámica periódicamente lee todos sus contenidos y se reescribe a sí misma. Por supuesto que esto la hace mucho más lenta que la SRAM.

En las PC modernas los módulos de memoria vienen en un formato conocido como DIMM (*Dual Inline Memory Module*), con distintas variantes de tamaño y se conectan directamente al *motherboard* mediante zócalos dedicados.

Almacenamiento no volátil

Periféricos

Fuente de alimentación

Ejercicios

1. Explicar en qué consiste el cuello de botella de Von Neumann. Nombrar una solución propuesta.
2. ¿En qué sentido *software* y *hardware* son equivalentes? ¿En cuál no?
3. Nombrar los tipos de memoria secundaria que conocen.
4. ¿Cuál es la diferencia entre una SRAM y una DRAM?
5. ¿Qué es el ciclo de instrucción? ¿Cuáles son sus pasos?
6. ¿A qué arquitectura corresponde una PC normal de escritorio de 1997, un Samsung Galaxy S2, una Raspberry Pi y una Mac de 1995?
7. Investigar el término *backplane* y contrastar con un *motherboard*.
8. ¿Cuál es el propósito de la jerarquía de memoria?
9. En el contexto de los microprocesadores, ¿qué significa la cantidad de bits? Por ejemplo, procesadores de 8 bits, de 32 bits, etc.
10. Dar una lista de los periféricos más comunes en una PC moderna y clasificarlos según sean de entrada o salida. ¿Hay periféricos que sean de entrada y salida a la vez?
11. De la fuente de alimentación se dice que tiene formato ATX y coincide con el *motherboard*. ¿Qué significa esta nomenclatura y de dónde viene?