

Actividad 1: Las computadoras y el sistema binario (Material complementario)

1. La Simplicidad que Impulsa la Complejidad

El texto base ya nos introdujo a la idea de que el sistema binario (0 y 1) es fundamental por su simplicidad. Pero, ¿por qué es tan importante esta simplicidad?

Imagina construir un edificio con solo dos tipos de ladrillos: uno que está "presente" y otro que está "ausente". Con solo esos dos estados, ¡puedes construir estructuras increíblemente complejas! Las computadoras hacen algo similar con la electricidad:

- **0 = Apagado (no hay voltaje o voltaje bajo)**
- **1 = Encendido (hay voltaje o voltaje alto)**

Esta dualidad es perfecta para los **transistores**, los componentes electrónicos minúsculos que actúan como interruptores dentro de los microprocesadores. Un transistor puede estar encendido o apagado, permitiendo o bloqueando el flujo de electricidad. Millones, incluso miles de millones de estos pequeños interruptores, trabajan juntos a una velocidad asombrosa para procesar información.

2. El Binario en Acción: Más Allá del Microprocesador

Si bien el microprocesador es el "cerebro" que interpreta las instrucciones binarias, la aplicación del sistema binario se extiende por cada rincón de la informática. Pensemos en algunos ejemplos más concretos:

2.1. Almacenamiento: Donde el 0 y el 1 Cobran Forma

- **Memorias RAM y ROM:** Piensa en la RAM como una pizarra donde la computadora anota lo que está usando en este momento. Cada "cuadradito" de esa pizarra es una celda de memoria que guarda un 0 o un 1. La ROM (Memoria de Solo Lectura) es más como un libro inalterable donde la computadora guarda sus instrucciones de arranque permanentes, también en forma de 0s y 1s.
- **Soportes Ópticos (CDs/DVDs/Blu-rays):** Cuando grabas un CD, un láser "quema" pequeños huecos (**pits**) en la superficie o deja áreas planas (**lands**). Al leerlo, otro láser detecta si hay un hueco (un 0) o no (un 1) por cómo se refleja la luz. Es una forma física de almacenar secuencias binarias.

- **Soportes Magnéticos (Discos Duros HDD, Cintas):** Aquí, el almacenamiento se basa en imantar diminutas partículas en una dirección (un 0) o en la dirección opuesta (un 1). Una cabeza lectora/escritora detecta estas polaridades para leer o escribir los datos binarios.

2.2. Comunicación: El Flujo Ininterrumpido de Datos Binarios

- **Buses de Comunicación:** Imagina los buses como "autopistas" dentro de tu computadora. Por estas autopistas viajan los datos binarios en forma de impulsos eléctricos. Cuantos más "carriles" tenga la autopista (más bits pueda transmitir el bus a la vez), más rápido se moverá la información.
- **Redes y Transmisión de Datos:** Cuando envías un mensaje por WhatsApp o navegas por una página web, tus datos se convierten en secuencias binarias. Estas secuencias viajan por cables de cobre como pulsos eléctricos, por fibra óptica como pulsos de luz, o por Wi-Fi como ondas de radio. Los protocolos de red, como TCP/IP, son como las "reglas de tráfico" que aseguran que esos 0s y 1s lleguen a su destino en el orden correcto y sin errores.

3. El Impacto del Binario en la Programación: Más Allá de los Números

Como programadores, aunque rara vez trabajamos directamente con 0s y 1s, es crucial entender que todo lo que escribimos se traduce a binario.

3.1. Tipos de Datos y su Representación Binaria

- **Enteros (int):**
 - **Recuerda:** Un entero de 8 bits puede representar $2^8 = 256$ valores diferentes. Si son sin signo, irían del 0 al 255. Si tienen signo, se usa el **complemento a dos**. Esta es una técnica ingeniosa para representar números negativos sin necesidad de un signo aparte, simplemente invirtiendo los bits y sumando uno.
 - **Ejemplo:** Para un entero de 4 bits:
 - 0000 = 0
 - 0001 = 1
 - 1111 = -1 (en complemento a dos)

- **Números en Punto Flotante (float, double):** Estos son más complejos. Imagina que quieres representar el número 3.14159. En binario, no hay un punto decimal. En su lugar, se usa una representación similar a la notación científica, donde un grupo de bits indica el signo, otro el exponente (la "potencia de 2" por la que se multiplica) y la **mantisa** (los dígitos significativos). Esto permite representar números muy grandes o muy pequeños con una precisión razonable.
- **Caracteres (ASCII, Unicode):**
 - **ASCII:** Cada letra, número o símbolo que ves en tu teclado tiene un código numérico asociado. Por ejemplo, la 'A' es 65 en decimal, que es 01000001 en binario. Cuando escribes 'A', la computadora en realidad procesa esa secuencia de bits.
 - **Unicode:** Es una expansión de ASCII para incluir caracteres de todos los idiomas del mundo. Mientras que ASCII usa 8 bits, Unicode puede usar hasta 32 bits por carácter, permitiendo representar millones de caracteres diferentes.
- **Colores (Modelo RGB):** Cuando ves un color en una pantalla, es una combinación de Rojo, Verde y Azul. Cada uno de estos colores se representa con un valor entre 0 y 255. En binario, esto significa 8 bits para cada color ($2^8 = 256$ valores).
 - **Ejemplo:** Un color como el verde lima fuerte (50, 205, 50) se traduce en binario como:
 - Rojo (50): 00110010
 - Verde (205): 11001101
 - Azul (50): 00110010

4. Unidades de Medida Binarias: Cuantificando el Mundo Digital

Es crucial distinguir entre las unidades de almacenamiento y las de velocidad de transferencia, aunque ambas se basan en el bit y el byte.

4.1. Almacenamiento: La Jerarquía de Bits y Bytes

Unidad	Equivalencia Binaria	Equivalencia (aproximada)	Decimal	Uso Común
Bit (b)	0 o 1	---		Unidad mínima de información.

Unidad	Equivalencia Binaria	Equivalencia (aproximada)	Decimal	Uso Común
Byte (B)	8 bits	---		Un carácter.
Kilobyte (KB)	1024 Bytes	≈ 1,000 Bytes		Documentos de texto pequeños, íconos.
Megabyte (MB)	1024 KB	≈ 1 millón de Bytes		Imágenes, canciones, documentos PDF.
Gigabyte (GB)	1024 MB	≈ 1 billón de Bytes		Películas, software, capacidad de RAM y USB.
Terabyte (TB)	1024 GB	≈ 1 trillón de Bytes		Discos duros de gran capacidad, almacenamiento en la nube.

¿Por qué 1024 y no 1000? En el sistema decimal, usamos múltiplos de 10 (10, 100, 1000). Pero en binario, es más natural usar potencias de 2 (2, 4, 8, 16, ..., 1024). Como $2^{10}=1024$, este número se convirtió en el estándar para las unidades binarias. Sin embargo, en el ámbito comercial, a menudo se usa la convención decimal (múltiplos de 1000) para discos duros, lo que puede causar pequeñas diferencias en la capacidad reportada.

4.2. Velocidades de Transferencia: Mbps vs. MBps

Es común confundir "Megabits por segundo" (Mbps o Mb/s) con "Megabytes por segundo" (MBps o MB/s).

- **Mbps (Megabits por segundo):** Se usa para medir la velocidad de las conexiones a internet. Si tu internet es de "100 Mbps", significa que puede transmitir 100 millones de bits por segundo.
- **MBps (Megabytes por segundo):** Se usa para medir la velocidad de transferencia de archivos, por ejemplo, al copiar un archivo de un disco duro a otro.

¡La clave es la 'b' minúscula para bits y la 'B' mayúscula para Bytes!

Para convertir de Mbps a MBps, simplemente divide por 8 (porque 1 Byte = 8 bits). Por ejemplo, una conexión de 100 Mbps equivale a $100/8=12.5$ MBps.

5. Práctica de Conversión:

La mejor forma de solidificar tu comprensión del sistema binario es practicar las conversiones.

5.1. Ejercicios Adicionales:

De Decimal a Binario:

1. Convierte 25 (decimal) a binario.
2. Convierte 42 (decimal) a binario.

De Binario a Decimal:

1. Convierte 10110 (binario) a decimal.
2. Convierte 10000000 (binario) a decimal.

5.2. Pistas y Consejos:

- **Para Decimal a Binario:** Siempre divide por 2 y anota el residuo. El truco es leer los residuos de abajo hacia arriba.
- **Para Binario a Decimal:** Recita las potencias de 2 de derecha a izquierda ($2^0=1, 2^1=2, 2^2=4, 2^3=8$, etc.) y suma solo aquellas donde el bit sea un '1'.

El sistema binario es la base silenciosa que permite que todo funcione en el mundo de la computación. Aunque a veces pueda parecer abstracto, su lógica simple es lo que lo hace tan increíblemente potente y fiable.