Representación y Almacenamiento de Datos

Organización de la memoria principal y de dispositivos de almacenamiento masivo. Representación de información como patrones de bits. Codificación de números. Errores de comunicación.

Organización de Memorias

Memoria Principal. Almacenamiento masivo: sistemas magnéticos y ópticos, memorias flash.

Memoria Principal

- La memoria principal de una computadora se organiza en unidades físicas llamadas **celdas**. El tamaño de las celdas de las primeras computadoras fue de 8 bits, y fue la razón por la que se designó a los patrones de este tamaño con el nombre propio **byte**: proviene de *bite* (en inglés "mordisco"), como la cantidad más pequeña de datos que un ordenador podía "morder" a la vez. Las celdas de las computadoras actuales suelen ser de 32 o 64 bits (4 u 8 bytes).
- Si se considera a una celda como una fila horizontal de bits, su mitad izquierda se denomina de orden alto, y su mitad derecha, de orden bajo. El bit del extremo izquierdo se dice es el más significativo, y el del extremo derecho, el menos significativo (por analogía con la representación de números).

Memoria Principal

- Las celdas de la memoria principal están numeradas secuencialmente desde 0, y sus números de orden, que sirven como identificadores, se denominan **direcciones**. Esto permite almacenar patrones de bits que excedan el tamaño de una celda (unidades lógicas de información), en varias celdas consecutivas.
- Además de los circuitos necesarios para representar bits, la memoria se compone de circuitos que permiten **leer** el contenido de una celda o **escribir** un nuevo patrón de bits en ella, proporcionando electrónicamente su dirección.
- La memoria se mide por su cantidad de bytes, independientemente del tamaño de sus celdas.

Sistema Internacional (decimal)		ISO/IEC 80000-13 (binario)	
Múltiplo (símbolo)	**Bytes**	**Múltiplo (símbolo)**	**Bytes**
Kilobyte (KB)	10^3	Kibibyte (KiB)	2^{10}
Megabyte (MB)	10^6	Mebibyte (MiB)	2^{20}
Gigabyte (GB)	10^9	Gibibyte (GiB)	2^{30}
Terabyte (TB)	10^{12}	Tebibyte (TiB)	2^{40}
Petabyte (PB)	10^{15}	Pebibyte (PiB)	2^{50}
Exabyte (EB)	10^{18}	Exbibyte (EiB)	2^{60}
Zettabyte (ZB)	10^{21}	Zebibyte (ZiB)	2^{70}
Yottabyte (YB)	10^{24}	Yobibyte (YiB)	2^{80}

Almacenamiento Masivo

- Debido a la volatilidad y al tamaño limitado de las memorias principales de las computadoras, éstas requieren dispositivos de memoria adicionales llamados **sistemas de almacenamiento masivo o secundario**, incluyendo discos magnéticos y memorias flash o de estado sólido.
- Las ventajas de estos sistemas sobre la memoria principal incluyen la no volatilidad o persistencia, grandes capacidades de almacenamiento, menores costos, y en muchos casos, la posibilidad de remover el medio de almacenamiento (pen drives) de la computadora para archivar información o hacerla portable (poder usarla en otra computadora).

Almacenamiento Masivo

- En general, cuando un dispositivo de almacenamiento masivo o cualquier periférico está conectado y disponible para una computadora sin necesidad de intervención humana, se dice que está “en línea” *(on-line)*, en cambio, si el dispositivo debe ser insertado en algún mecanismo o encendido para que la computadora puede acceder a él, se dice que está “fuera de línea” *(off-line)*.
- La principal desventaja de los sistemas de almacenamiento masivo es que, excepto las memorias flash, requieren movimientos mecánicos y, en consecuencia, mucho más tiempo para almacenar o recuperar información que la memoria principal, donde todas las actividades se realizan electrónicamente.

Almacenamiento Masivo

- En los sistemas de almacenamiento masivo, la información se agrupa en grandes unidades llamadas **archivos**. Un archivo típico puede consistir de un documento de texto completo, una fotografía, un video, un programa, una grabación musical o un conjunto de datos sobre empleados de una compañía.
- Los archivos a menudo tienen divisiones naturales determinadas por la información representada; por ejemplo, un archivo de texto suele dividirse en líneas, párrafos o páginas, mientras que un archivo de empleados de una compañía contendrá múltiples unidades, cada una consistente de información sobre un empleado. Estas unidades conceptuales de información se denominan **registros lógicos**.

Almacenamiento Masivo

- Los registros lógicos a menudo consisten de unidades más pequeñas que representan características o atributos de las unidades de información representadas en los registros, llamadas **campos**; por ejemplo, un registro de empleado probablemente consistiría de campos para la CUIL, nombre, dirección, etc. En general, los registros lógicos dentro de un archivo se identifican unívocamente mediante un campo particular dentro del registro, llamado **campo clave**, mientras que el valor de ese campo se denomina **clave** del registro; por ejemplo, para un registro de empleado el campo clave sería la CUIL (Clave Única de Identificación Laboral).
- Sin embargo, las unidades físicas de lectura o escritura en los sistemas de almacenamiento masivo dependen de las características de los dispositivos y se denominan **registros físicos**.

Almacenamiento Masivo

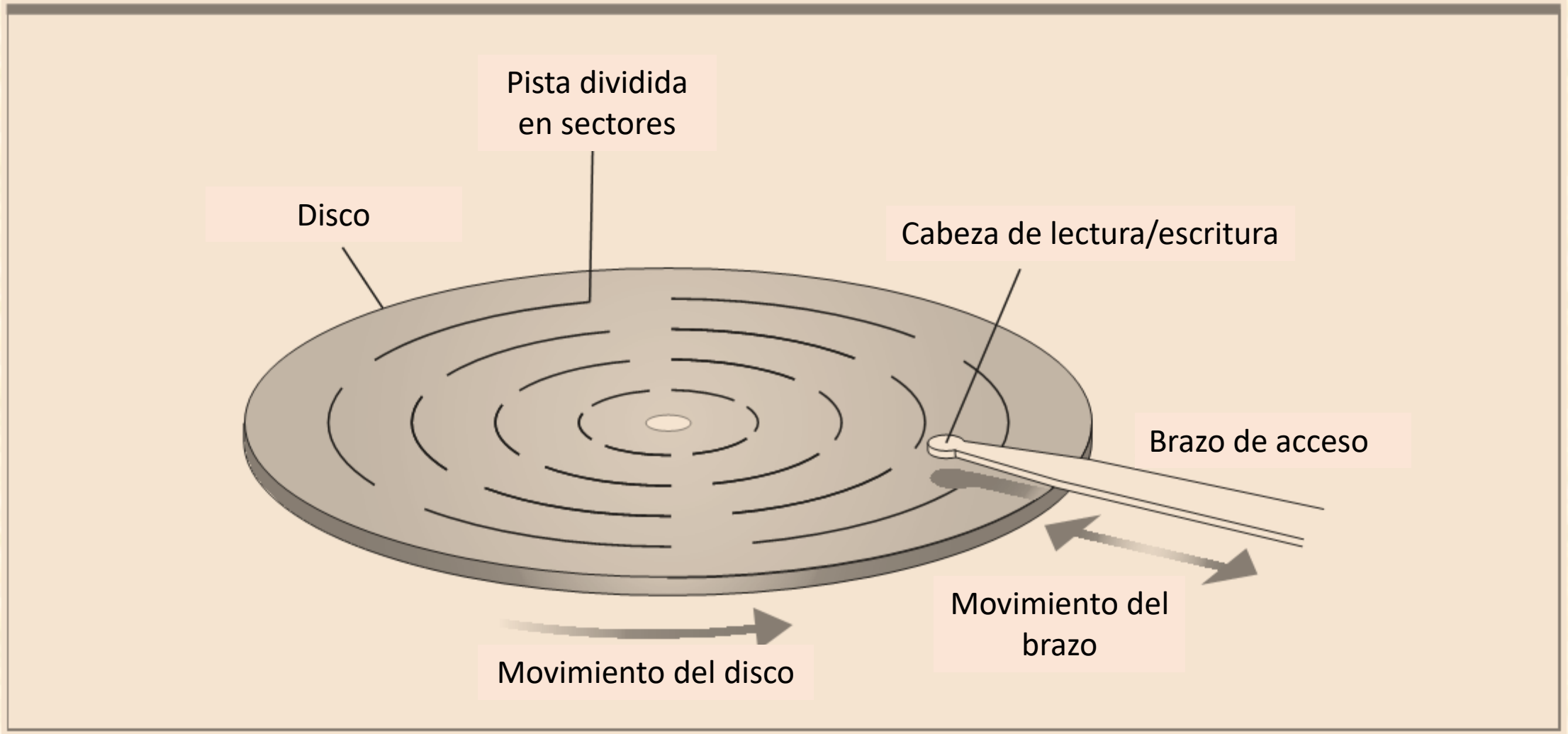
- Los tamaños de registros lógicos raramente coinciden con los tamaños de registros físicos, por lo que en general, algunos registros lógicos pueden estar partidos en dos o más registros físicos. Una solución general a este problema es establecer un área en la memoria principal lo suficientemente grande como para contener varios registros físicos llamada **buffer**, y usarla para recomponer registros lógicos.
- En general, un buffer es un área de almacenamiento temporal que se usa para transferir datos de un dispositivo a otro. Por ejemplo, las impresoras actuales contienen memoria propia, gran parte de la cual se usa para mantener porciones de documentos recibidas que aún no se imprimieron.

Sistemas Magnéticos

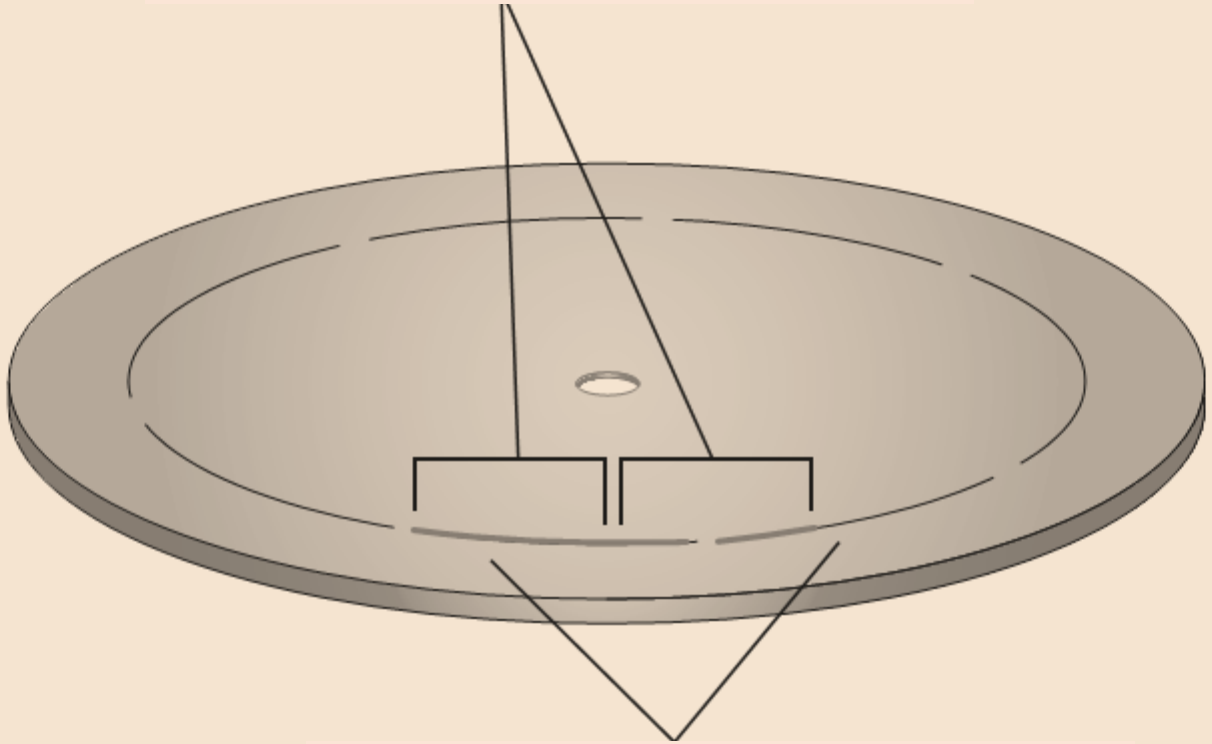
- Actualmente, los sistemas de almacenamiento en **discos magnéticos** dominan el campo del almacenamiento masivo en línea. Consisten de varios discos giratorios paralelos (en general 3 a 6) con revestimiento magnético montados en un perno central, y a distancia suficiente para que un brazo con cabezales paralelos de lectura/escritura para cada superficie se deslice entre ellos sin tocarlas.
- Las circunferencias que determina un cabezal de lectura/escritura sobre un disco se denominan **pistas** *(tracks)*, y las pistas que comprende el conjunto de cabezales de lectura/escritura en sus distintas posiciones de desplazamiento, **cilindros** *(cylinders)*.

Sistemas Magnéticos

- Cada pista de un sistema de discos se divide en arcos con la misma capacidad de almacenamiento (típicamente 512 bytes), llamados **sectores**. Los sectores son las unidades físicas de almacenamiento y recuperación de información, o registros físicos.
- En los sistemas de alta capacidad de almacenamiento se suelen agrupar pistas adyacentes en aproximadamente diez **zonas**, de modo que las pistas de zonas más próximas al centro del dispositivo tienen muchos menos sectores que las pistas de zonas menos centrales. De todos modos en cada zona, los bits en sectores de pistas más próximas al centro del disco se almacenan más comprimidos que los de sectores de pistas más alejadas del centro, debido a la diferencia de longitudes de arcos.


Pista dividida
en sectores
Disco
Cabeza de lectura/escritura
Brazo de acceso
Movimiento del
brazo
Movimiento del disco

Los registros lógicos corresponden a divisiones naturales de los datos



Los registros físicos corresponden al tamaño de un sector

Sistemas Magnéticos

- Las localizaciones de pistas y sectores se establecen magnéticamente mediante el proceso de **formateo** o inicialización del dispositivo, que reserva varios bits entre sector y sector para registrar sus números relativos dentro de la pista y un patrón para la detección de errores llamado **CRC** (Control de Redundancia Cíclica), que consiste en el O EXCLUSIVO entre los bytes de cada sector.
- El direccionamiento de sectores dentro de un dispositivo requiere la precisión de un número de cilindro y otro de superficie para identificar a una pista, y el número de sector dentro de esa pista.

Sistemas Magnéticos

- El rendimiento o **performance** de un sistema de discos se evalúa por medio de
  - El **tiempo de acceso** *(access time)*, que es la suma de
    - El **tiempo de búsqueda** *(seek time)*: necesario para mover las cabezas de lectura escritura de un cilindro a otro
    - El tiempo de latencia *(latency time)* o demora rotacional *(rotation delay)*: requerido para que el comienzo del sector a leer o escribir llegue a la cabeza de lectura/escritura, que en promedio es de media rotación
  - La **tasa de transferencia** *(transfer rate)*: relación entre la velocidad de rotación y la longitud del sector (variable según la zona de la pista). Los sistemas de discos son capaces de rotar miles de veces por segundo, con tasas de transferencia que se miden en MiB/seg.

Memoria Principal vs Sistemas Magnéticos

Memoria Principal

- Acceso a la información mediante circuitos electrónicos
- Tiempos de acceso medidos en nanosegundos (milmillonésima parte de un segundo)

Sistemas Magnéticos

- El acceso a la información requiere movimientos físicos
- Tiempos de búsqueda, latencia y acceso medidos en milisegundos (milésima parte de un segundo)

El tiempo de espera para recuperar información de un disco puede parecer una eternidad para un circuito electrónico esperando un resultado.

Memorias Flash

- La tecnología de memoria flash no requiere de movimientos mecánicos para el almacenamiento o recuperación de información, lo cual implica que sus sistemas tengan tiempos de acceso significativamente menores que los magnéticos y ópticos. Además sus dispositivos son pequeños e insensibles a shocks físicos, en contraste con los de memoria magnética u óptica, por lo que su portabilidad resulta muy práctica.
- En un sistema de memoria flash los bits se almacenan enviando señales electrónicas directamente al medio de almacenamiento, causando que los electrones queden atrapados en pequeñas cámaras de dióxido de silicio, alterando así las características de pequeños circuitos electrónicos. Como estas cámaras son capaces de mantener sus electrones cautivos por varios años, esta tecnología es apropiada para el almacenamiento de datos fuera de línea *(off-line)*.

Memorias Flash

- La desventaja de los sistemas de memoria flash es que para grabar nueva información deben liberarse los electrones atrapados en las cámaras de dióxido de silicio mediante un proceso de borrado, que al repetirse las va erosionando lentamente. Esto hace que aunque la información pueda grabarse en pequeñas unidades dimensionadas en bytes como en la memoria principal, estos sistemas no sean apropiados para su uso en esta jerarquía de almacenamiento, ya que sus contenidos deberían ser alterados varias veces por segundo.
- Las memorias flash son adecuadas para aplicaciones cuyos cambios en la memoria se realicen con poca frecuencia (cámaras digitales, teléfonos celulares, agendas digitales de bolsillo, dispositivos USB –*Universal Serial Bus*). Sus dispositivos tienen capacidades de hasta cientos de GiB.

Representación de Información como Patrones de Bits

Representación de texto, valores numéricos, imágenes y sonido.

Representación de Texto

- La información en forma de texto normalmente se representa mediante un código en el que a cada uno de los diferentes símbolos en un texto, como letras del alfabeto y signos de puntuación, se le asigna un único patrón de bits.
- Los estándares actuales son el **ASCII** *(American Standard Code for Information Interchange)*, que fue establecido por el ANSI *(American National Standards Institute)* y emplea patrones de un byte para representar letras en mayúscula y en minúscula, signos de puntuación, dígitos decimales y ciertos patrones de control como "salto de línea" *(line feed)*, "retorno de carro" *(carriage return)*, "tabulación" *(tab)* y "fin de archivo" *(end of file)*.

ASCII

Lista parcial del código ASCII, en el cual los símbolos se representan en patrones de 7 bits, pero se extienden con un 0 a la izquierda para obtener patrones de un byte, más adecuados para las computadoras.

La International Organization for Standardization (conocida como ISO, en referencia a la palabra griega *isos*, que significa igual), desarrolló varias extensiones para comprender grupos de lenguajes, agregando patrones con un 1 en el bit más significativo. Por ejemplo, un estándar provee los símbolos necesarios para representar texto en la mayoría de los idiomas de Europa Occidental. Incluidos entre estos 128 patrones adicionales se encuentran las representaciones de ñ, Ñ, €, £, vocales con acentos graves y circunflexos (francés) o con diéresis (alemán), etc.

Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex
line feed	00001010	0A	>	00111110	3E	^	01011110	5E
carriage return	00001011	0B	?	00111111	3F	_	01011111	5F
space	00100000	20	@	01000000	40	`	01100000	60
!	00100001	21	A	01000001	41	a	01100001	61
”	00100010	22	B	01000010	42	b	01100010	62
#	00100011	23	C	01000011	43	c	01100011	63
$	00100100	24	D	01000100	44	d	01100100	64
%	00100101	25	E	01000101	45	e	01100101	65
&	00100110	26	F	01000110	46	f	01100110	66
’	00100111	27	G	01000111	47	g	01100111	67
(	00101000	28	H	01001000	48	h	01101000	68
)	00101001	29	I	01001001	49	i	01101001	69
*	00101010	2A	J	01001010	4A	j	01101010	6A
+	00101011	2B	K	01001011	4B	k	01101011	6B
’	00101100	2C	L	01001100	4C	l	01101100	6C
-	00101101	2D	M	01001101	4D	m	01101101	6D
.	00101110	2E	N	01001110	4E	n	01101110	6E
/	00111111	2F	O	01001111	4F	o	01101111	6F
0	00110000	30	P	01010000	50	p	01110000	70
1	00110001	31	Q	01010001	51	q	01110001	71
2	00110010	32	R	01010010	52	r	01110010	72
3	00110011	33	S	01010011	53	s	01110011	73
4	00110100	34	T	01010100	54	t	01110100	74
5	00110101	35	U	01010101	55	u	01110101	75
6	00110110	36	V	01010110	56	v	01110110	76
7	00110111	37	W	01010111	57	w	01110111	77
8	00111000	38	X	01011000	58	x	01111000	78
9	00111001	39	Y	01011001	59	y	01111001	79
:	00111010	3A	Z	01011010	5A	z	01111010	7A
;	00111011	3B	[	01011011	5B	{	01111011	7B
<	00111100	3C	\	01011100	5C	|	01111100	7C
=	00111101	3D	]	01011101	5D	}	01111101	7D

Representación de Texto

- Las distintas extensiones del ASCII ocasionan problemas de compatibilidad para visualizar texto con aplicaciones que usan extensiones diferentes, y acotan la representación de texto al grupo de lenguajes que soporta cada extensión. Además, muchos lenguajes de Asia y Europa Oriental tienen más de 256 caracteres, por lo que no pueden cubrirse con ninguna extensión.
- Varias empresas líderes en la manufactura de hardware y software cooperaron en el desarrollo de **Unicode**, que usa patrones de 16 bits para representar cada símbolo. Con 65536 patrones de bits diferentes se puede escribir texto en chino, japonés o hebreo.

Representación de Texto

- Un archivo consistente de una larga secuencia de símbolos codificados usando ASCII o Unicode se denomina **archivo de texto**.
- Los archivos de texto simple o plano se manipulan con programas utilitarios denominados **editores de texto** *(text editors)*, y los archivos de texto con formato sólo se pueden editar con programas denominados **procesadores de palabras** *(word processors)*.
- Los archivos producidos por un procesador de palabras contienen códigos exclusivos de la aplicación para representar cambios de fuente, de alineación, sangrado e interlineado de párrafos, tamaños y características de fuentes (resaltada, cursiva, subrayada, etc.).

Representación de Valores Numéricos

- Obsérvese que si, por ejemplo, se sumaran los códigos ASCII de los dígitos 0 y 1 (en hexadecimal 30+31), el resultado sería el código correspondiente a la letra “a” (61), y no al dígito 1. Las operaciones aritméticas son inviables usando la codificación de ASCII o Unicode.
- Para que la Unidad Aritmético-Lógica pueda realizar las operaciones de su competencia, los números deben representarse con una codificación binaria que sea más compatible con el diseño de circuitos digitales.
- Para representar números enteros, tanto positivos como negativos, se usa una codificación llamada en **complemento a 2**, y para representar números fraccionarios, otra que se denomina en **punto flotante**, que se verán en detalle más adelante.

Representación de Imágenes
Pixeles

- Una forma de representación de imágenes es interpretarlas como una colección de puntos, cada uno de los cuales se denomina **pixel** (del inglés ***pi**cture **el**ement*), y codificar la apariencia de cada pixel. Una secuencia de pixeles codificados se denomina **mapa de bits** *(bit map)*. Las impresoras y pantallas operan en función de pixeles.
- Para imágenes en blanco y negro, cada pixel se representa con un bit, y para escalas de gris y color se requieren varios bits por pixel. Para el caso de imágenes en color, existen otros esquemas de codificación de pixeles, siendo los más comunes **RGB** (*Red-Green-Blue*, por rojo, verde y azul) y **luminancia-cromatismo** (transmisión de televisión en color).

Imagen bmp de 1 bit por pixel (blanco y negro)

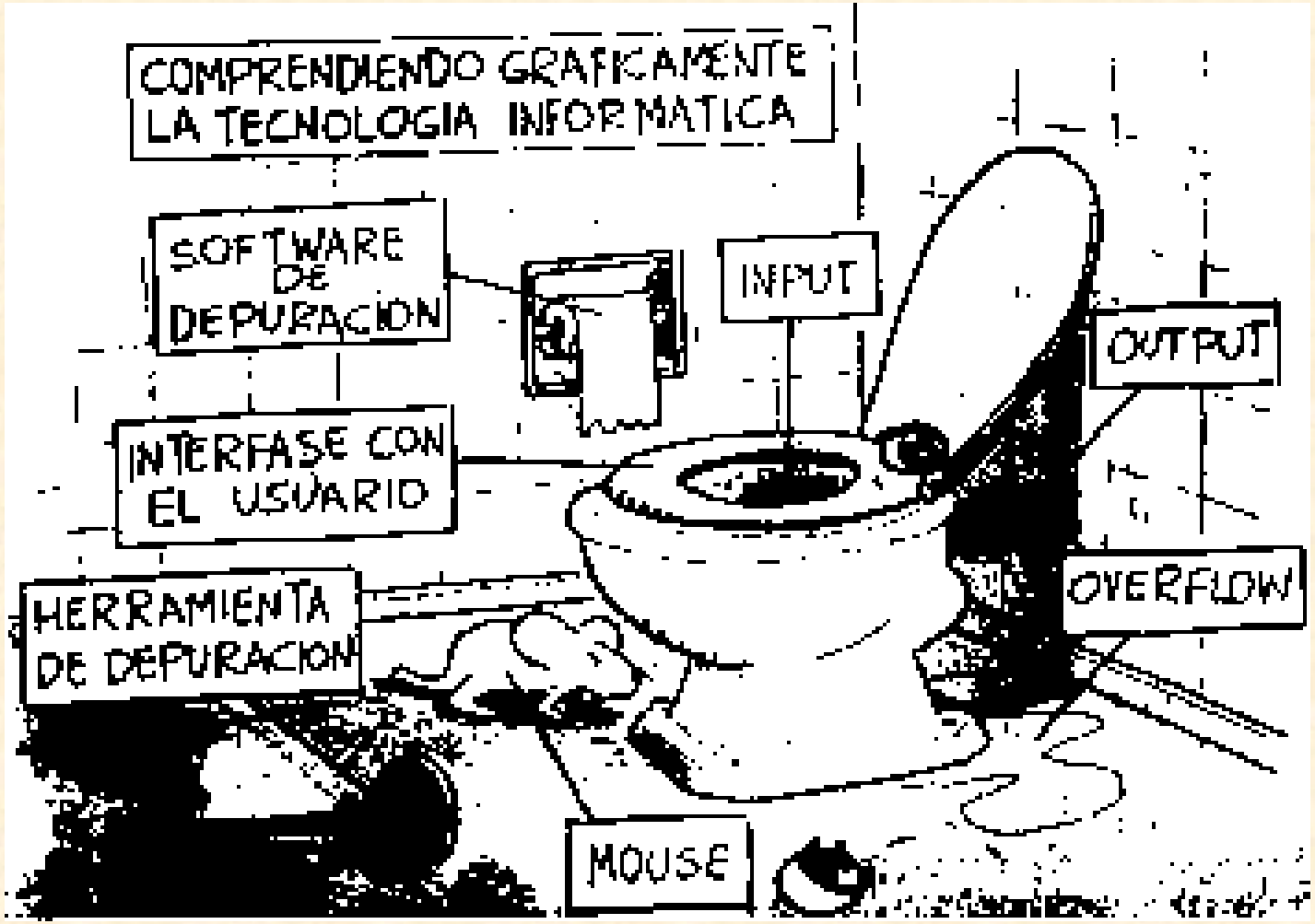


Imagen bmp de 4 bits por pixel (16 colores)

Imagen bmp de 8 bits por pixel (256 colores)

Imagen bmp de 24 bits por pixel

Representación de Imágenes
Pixeles

- En el sistema RGB se usan tres bytes por pixel, para codificar la intensidad de cada color primario.
- El sistema de luminancia-cromatismo también tiene tres componentes para cada pixel, pero el primero es para el brillo o luminancia del pixel, y los otros dos para las diferencias entre la luminancia y las cantidades de luz azul y roja (cromatismos azul y rojo).
- La cantidad de puntos por pulgada (ppp) o dpi *(dots per inch)* determina la **definición** de una imagen. El problema de los mapas de bits es la escalabilidad o zoom digital, que al ampliar imágenes de baja definición produce efectos de granularidad.

Imagen de 200 dpi (definición normal) achicada

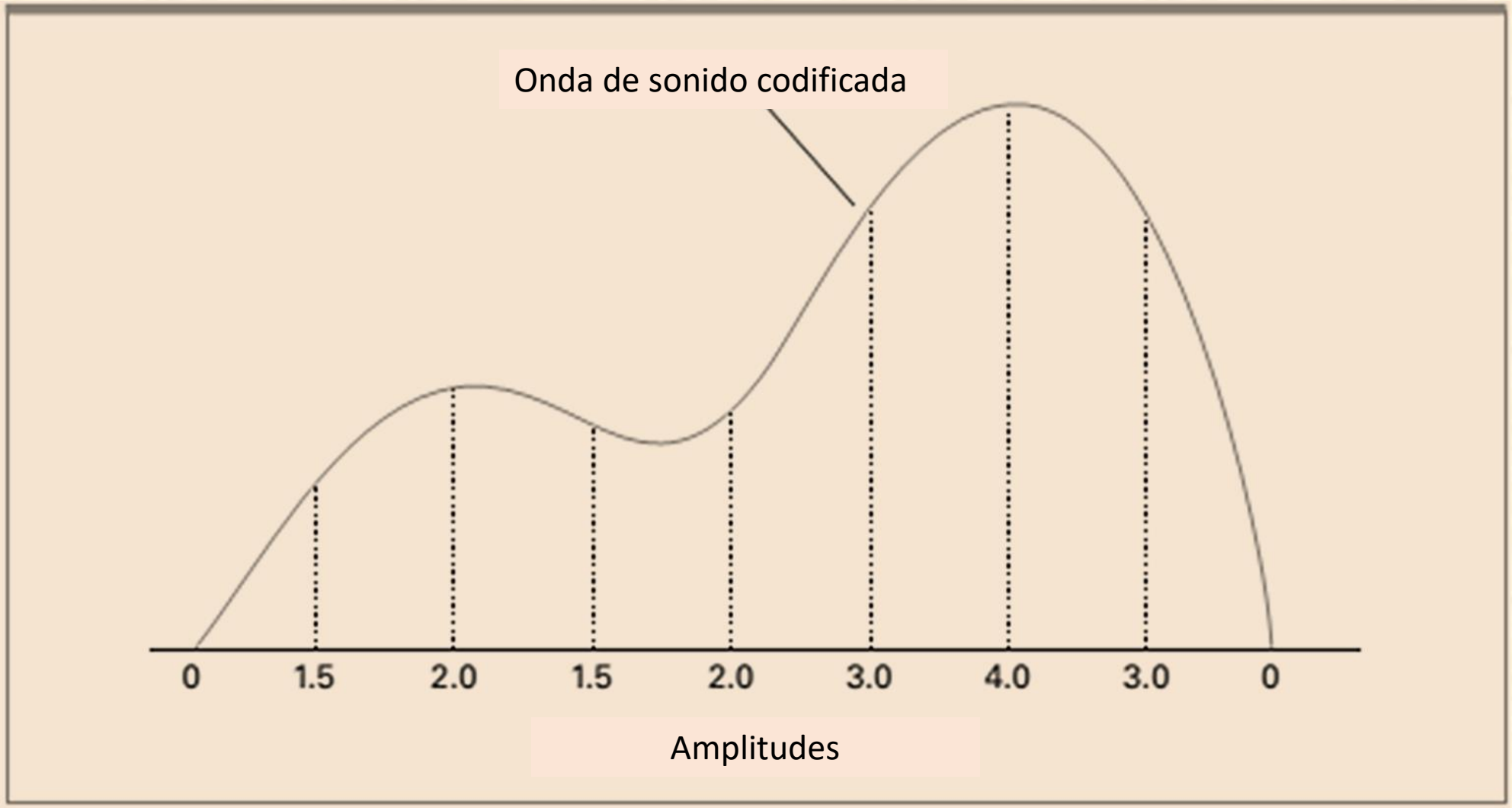
Imagen de 75 dpi (baja definición) agrandada

Representación de Imágenes
Técnicas de Geometría Analítica

- Otra alternativa para representar imágenes, que evita los problemas de escalabilidad, consiste en describirlas como una colección de estructuras geométricas codificadas con técnicas de geometría analítica.
- Esta forma de representación se utiliza en aplicaciones CAD (*computer-aided design*, diseño asistido por computadora), de dibujo, y para codificar caracteres de fuentes escalables *(TrueType)* en sistemas de procesamiento de palabras.

Representación de Sonido
Muestreo

- El método más genérico de codificar información de audio consiste en registrar la amplitud de la onda de sonido a intervalos regulares.
- La frecuencia de registros, también llamada **velocidad de muestreo**, determina la **fidelidad** del sonido. Para los CD de audio actuales, la velocidad de muestreo *(sample rate)* es de 44100 muestras por segundo.
- Cada muestra *(sample)* se representa en 16 (mono) o 32 (estéreo) bits. Cada segundo de música grabada en estéreo ocupa más de un millón de bits.


Onda de sonido codificada
0
1.5
2.0
1.5
2.0
3.0
4.0
3.0
0
Amplitudes

Representación de Sonido
MIDI

- Otro sistema de codificación es el **MIDI** *(Musical Instrument Digital Interface)*, que se usa para sintetizadores de música basados en teclados electrónicos, en música de videojuegos y efectos de sonido de sitios web.
- Este sistema codifica órdenes para producir sonido en un sintetizador en vez de codificar al sonido mismo, en términos de qué instrumento ejecuta qué nota musical por cuánto tiempo. Por ejemplo que un clarinete ejecute un do durante dos segundos se puede codificar en 3 bytes, en lugar de los más de dos millones de bits que implicaría un muestreo.

Codificación de Números

Enteros en complemento a 2 y en exceso. Fraccionarios en punto flotante.

Representación de Enteros en Complemento a 2

- Es el sistema más común para representar valores enteros en las computadoras actuales, y se limita a usar la cantidad de bits que determine el tamaño de los registros de la CPU (normalmente 32 o 64 bits).
- El bit más significativo determina el signo del valor representado. Los positivos se representan comenzando con una cadena de ceros, y luego contando progresivamente en binario hasta alcanzar un patrón con un cero seguido de unos. Los negativos se obtienen comenzando con una cadena de unos, y luego contando regresivamente hasta alcanzar un patrón con un uno seguido de ceros.

Sistemas de Notación en Complemento a 2

Para estudiar las propiedades de los sistemas de complemento a 2, se ejemplifican usando 3 y 4 bits. Las representaciones de valores positivos y negativos de la misma magnitud son idénticas de derecha a izquierda hasta el primer 1, y los patrones restantes son complementos a uno, el uno del otro.
El **complemento a 1** de un patrón es el patrón que se obtiene cambiando ceros por unos, y unos por ceros.

Patrones de 3 bits

Patrón Binario	Valor Decimal
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

Patrones de 4 bits

Patrón Binario	Valor Decimal	Patrón Binario	Valor Decimal
0111	7	1111	-1
0110	6	1110	-2
0101	5	1101	-3
0100	4	1100	-4
0011	3	1011	-5
0010	2	1010	-6
0001	1	1001	-7
0000	0	1000	-8

Representación de Enteros en Complemento a 2

- Para **decodificar** un patrón con bit de signo 0, se interpreta en notación binaria, y si el bit de signo es 1, se determina su magnitud con el procedimiento de copiar de derecha a izquierda hasta el primer 1 inclusive y complementar a 1 el resto (otra alternativa es complementar a 1 todo el patrón y luego sumarle 1).
- La **suma en complemento 2** se realiza igual que la suma binaria, truncando el bit más significativo si hay un acarreo que exceda la longitud del patrón de representación.
- Cuando un suma resulta en un valor que excede el rango de valores que pueden representarse, se produce un **error de desborde** *(overflow)*.

Sumas en Complemento a 2

Problema en base 10	Problema en complemento a 2	Observaciones
3 + 2 5	0011 + 0010 0101	Suma de positivos positiva
-3 + -2 -5	1101 + 1110 ~~1~~1011	Suma de negativos negativa; se trunca acarreo
7 + -5 2	0111 + 1011 ~~1~~0010	Suma de valores con signos opuestos (puede resultar cualquier signo); se trunca acarreo
5 + 4 9	0101 + 0100 **1**001	Desborde (suma de valores con igual signo, con resultado de signo opuesto)

Representación de Enteros en Exceso

- Cada uno de los valores en un sistema de notación en exceso se representa con un patrón de bits de igual longitud.
- Considerando todos los patrones representables de la longitud establecida ordenados en numeración binaria, se toma el patrón con un 1 seguido de ceros como el valor 0 de la representación; los patrones siguientes representarán 1, 2, 3, … y los precedentes –1, –2, –3, … (los bits de signo son inversos al los del sistema de complemento a 2).
- Un sistema de notación en exceso en n bits se denomina notación en exceso de 2^{n-1}, puesto que para decodificar un patrón, a su equivalente decimal se le resta 2^{n-1}.

Sistemas de Notación en Exceso

En 3 bits: exceso de $2^2 = 4$

Patrón Binario	Valor Decimal
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4

En 4 bits: exceso de $2^3 = 8$

Patrón Binario	Valor Decimal
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0

Patrón Binario	Valor Decimal
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

Representación de Fraccionarios en Punto Flotante

- Cada uno de los valores en un sistema de notación en punto flotante se representa con un patrón de bits de igual longitud.
- El primer bit de cada patrón representa el signo (el exponente al que elevar a -1 para obtener un factor que determine el signo): 1 para negativos y 0 para no negativos. Los bits restantes se dividen en dos grupos o campos, para representar el exponente en notación en exceso, y la mantisa. Para patrones de 32 bits se usan 8 bits para exponente y 23 para la mantisa (punto flotante de precisión simple -*Single Precision Floating Point)*.
- Un sistema con patrones de 8 bits, por ejemplo, asignaría 3 bits para el exponente, y 4 para la mantisa: `seeemmmm` donde `s`, `e` y `m` representan cualquier dígito binario. En notación científica binaria esto equivale a $0.mmmm \times 10^{eee-100}$ con el signo que represente `s`.

Representación de Fraccionarios en Punto Flotante

- Para evitar representaciones múltiples de un mismo valor, se impone que el primer bit de la mantisa siempre sea 1, excepto para la representación del valor 0, que por convención, se define con el patrón de bits de todos ceros. Las representaciones que cumplen esta regla se dice que están en **forma normalizada**.
- Como en las formas normalizadas el primer bit de la mantisa siempre es `1`, se puede extender la mantisa un bit adicional ocultando la representación de ese `1`: en notación científica binaria `seeemmmm` representaría $1.mmmm \times 10^{eee-100}$ con el signo que represente `s`.

Representación de Fraccionarios en Punto Flotante

- Los errores que se producen al perder los dígitos menos significativos de una mantisa por límites de representación, se denominan **errores de truncamiento**.
- Por ejemplo, para representar el binario 10.101 según la primer alternativa de normalización, primero se expresa en notación científica como 0.10101×10^{10}, luego se suma 100 al exponente para obtener su representación en exceso: $0|110|10101$. La segunda alternativa de normalización gana un bit extra de precisión: $100.101 = 1.0101 \times 10^{1} \equiv 0|101|0101$.

Errores de Comunicación

Detección de errores con bits de paridad. Corrección de Errores.

Bits de Paridad

- Un método para detectar errores de transmisión de datos entre dispositivos es agregar un bit de control de manera que los patrones que se almacenen y transmitan tengan una cantidad par de unos. Este bit extra se denomina **bit de paridad**. Es común que las celdas de la memoria principal tengan un bit extra para este propósito. Este método tiene la limitación de que si se produce un error que invierte una cantidad par de bits, el bit de paridad no indicaría error.
- Para patrones de bits grandes, como los de registros físicos de sistemas de discos magnéticos, se suele acompañar a cada patrón con una colección de bits de paridad conformando un **checkbyte**.