

## Representación de Datos en la Computadora

Básicamente se necesita representar en la computadora dos tipos de datos: texto y números.

Pero estos datos han de representarse internamente por medio de secuencias de impulsos eléctricos ya que, en definitiva, eso es lo que puede manipular la computadora.

Por tanto, los símbolos tipográficos y las cantidades numéricas han de tener una representación en base a impulsos eléctricos.

Un impulso eléctrico lo representamos por medio de un 1 (representación lógica): hay presencia de impulso. Y a través de 0, hay ausencia de impulso.

Estos son los únicos dos símbolos utilizados, por ello se dice que es un sistema de representación binaria.

Fuera del ámbito de las computadoras actuales también ha habido muchos intentos para representar símbolos y lograr la comunicación entre las personas:

- Pictogramas.
- Alfabeto y números indo-arábigos.
- Lenguaje de señas, para comunicación con personas que carecen del sentido del oído.
- Señales de tránsito.

- Clave Morse:

a .-	g --.	m --	s ...	y -.--
b - ...	h ....	n -.	t -	z --..
c -..	i ..	o ---	u ..-	
d -..	j .---	p .--.	v ...-	
e .	k -.-	q --.-	w .--	
f ..-	l .-..	r .-.	x -.-	

Tomando como ejemplo esta última, se puede decir lo siguiente:

- Repertorio de símbolos limitado: . y -
- Representación del alfabeto con combinaciones de . y - en cantidades variables.
- Necesita de la intervención humana directa en el envío y la recepción de datos.
- Realización de pausas más largas entre la emisión de un carácter y el siguiente.

Por esta última razón se complementa mucho con el criterio humano.

En la computadora se necesita un tipo de representación que no dé lugar a interpretaciones ambiguas o erróneas, sino únicas y acertadas. La solución es utilizar una secuencia de dígitos binarios de igual tamaño para cada símbolo y que el espacio entre un símbolo y otro sea otro símbolo de igual tamaño.

Han habido muchos intentos a lo largo de la historia de la computación para fabricar tablas de equivalencias de símbolos tipográficos y secuencias de dígitos binarios, el más difundido es el código **ASCII** (**American Standar Code for Information Interchange**), que utilizó 7 dígitos en su primera versión y utiliza 8 actualmente.

La cantidad de dígitos de esa secuencia dependerá de la cantidad de símbolos que se quiera representar. Por ejemplo:

Con 2 casilla se pueden realizar las combinaciones: 00, 01, 10 y 11. En total, 4 combinaciones distintas. Si cada combinación representa a un símbolo tipográfico, se representarán un máximo de 4 símbolos tipográficos.

Con 3 casilla se pueden realizar las combinaciones: 000, 001, 010, 011, 100, 101, 110, 111. En total, 8 combinaciones distintas. Si cada combinación representa a un símbolo tipográfico, se representarán un máximo de 8 símbolos tipográficos.

En general, con n casillas y dos dígitos para realizar combinaciones, se pueden representar  $2^n$  combinaciones posibles.

El código ASCII, que utilizó 7 dígitos en su primera versión, podía representar  $2^7$  (= 128) símbolos. Al agregar un bit más se pudo representar un total de  $2^8$  (= 256) símbolos.

Buscar en Internet la tabla ASCII e identificar: códigos de control, tipos numéricos, letras del alfabeto en mayúsculas, letras del alfabeto en minúsculas, y otros símbolos intercalados. De la posición 129 en adelante, numeradas 128 a 255, se tienen tipos que no son del alfabeto inglés, símbolos para colocación de rellenos y trazado de cuadrículas, etc.

Para establecer sentencias de texto, que no son poco utilizadas en computación (bases de datos con información textual de clientes, artículos, empleados, etc.; mensajería; documentos; etc.) se realiza la sustitución del carácter por la secuencia binaria correspondiente. Ejemplos:

Te amo:	01010100	011100101	001000000	01100010	01101101	01101111
---------	----------	-----------	-----------	----------	----------	----------

Para la transmisión segura de datos se utilizan, entre otros, procedimientos de verificación y detección de errores como la paridad para y la paridad impar.

Paridad Par: A cada grupo de ocho bits, de la hilera, se le agrega un bit adicional a la izquierda. Este será un uno si la cantidad de unos es impar, y cero si la cantidad de unos es par, de tal manera que la cantidad de unos del nuevo grupo de nueve bits sea par.

Paridad Impar: A cada grupo de ocho bits, de la hilera, se le agrega un bit adicional a la izquierda. Este será un cero si la cantidad de unos es impar, y uno si la cantidad de unos es par, de tal manera que la cantidad de unos del nuevo grupo de nueve bits sea impar.

Ejemplo:

Codificar la siguiente frase utilizando paridad Impar:

*Coco rallado:*

C	o	c	o		r	a	l
001000011	101101111	101100011	101101111	000100000	101110010	001100001	101101100
l	a	d	o				
101101100	001100001	001100100	101101111				

## Representación de Cantidades Numéricas

### Sistemas de Numeración Posicional

Las personas utilizamos los números en muchas actividades de la vida diaria: para medir el tiempo horario (sistema en base 60), para calcular las fechas, para indicar nuestra edad, para nuestros gastos (dinero), para medir nuestro desempeño académico, para medir la temperatura y presión atmosféricas, para medir nuestro estado de salud (temperatura corporal, presión arterial, etc.), para medir el estado de nuestra visión, para medir distancias, para la realización de cálculos matemáticos complejos, etc. Es imposible escapar de los números.

Existen dos formas de representar los valores numéricos de las cantidades:

- a) Analógica.
- b) Digital.

Representación de cantidades Analógicas: una cantidad se denota por otra que es directamente proporcional a ella, ejemplo: termómetro, velocímetro, balanza (la del peso y la de aguja), voltímetro, reloj de agujas, medidor de gasolina, la cinta métrica, la regla para medir yardas en las sastrerías, etc. Características de estas cantidades: a) intervalo continuo de valores, b) el valor se presta a interpretación y es aproximado.

Representación de cantidades Digitales: Las cantidades se representan por símbolos denominados dígitos. Características: a) la representación de cantidades varía en etapas discretas, b) no hay ambigüedad al leer las cantidades. Ejemplos: reloj digital, ábaco, caja registradora, calculadoras, computadoras, etc.

### Sistemas de Números Digitales

Los más comunes: Decimal, Octal, Binario y Hexadecimal. Se componen de los siguientes dígitos:

Decimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Binario: 0, 1.

Octal: 0, 1, 2, 3, 4, 5, 6, 7.

Hexadecimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Todos ellos son sistemas de valor posicional, ya que el valor de un dígito depende de su posición.

Ejemplos:

**34:**

Dígito de mayor valor: 3 (3 decenas)

Dígito de menor valor: 4 (4 unidades)

$$34 = 3 * 10 + 4 * 1 = 30 + 4 = 34$$

**247:**

2: porta las centenas

4: porta las decenas

7: porta las unidades

$$247 = 2 * 100 + 4 * 10 + 7 * 1 = 200 + 40 + 7$$

**625.41:**

6 centenas:  $6 * 100: 6 * 10^2$

2 decenas:  $2 * 10: 2 * 10^1$

5 unidades:  $5 * 1: 5 * 10^0$

4 décimos:  $4 * (1/10): 4 * 10^{-1}$

1 centésimo:  $1 * (1/100): 1 * 10^{-2}$

Observar:

- Al considerar 1 espacio decimal se pueden contar 10 números diferentes, comenzando con cero.
- Al considerar 2 espacios decimales se pueden contar 100 números diferentes, comenzando con cero.
- En general, al considerar N espacios decimales se pueden contar  $10^N$  números diferentes, comenzando con cero.
- El número mayor siempre será  $10^N - 1$ .
- Estas consideraciones se extienden para cualquier base. En general se tendrá, para una cantidad de dígitos determinada:

$\text{base}^N$  números representables y será  $\text{base}^N - 1$  el número mayor.

**Conteo en sistemas de numeración posicional**

Decimal	Binario	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F