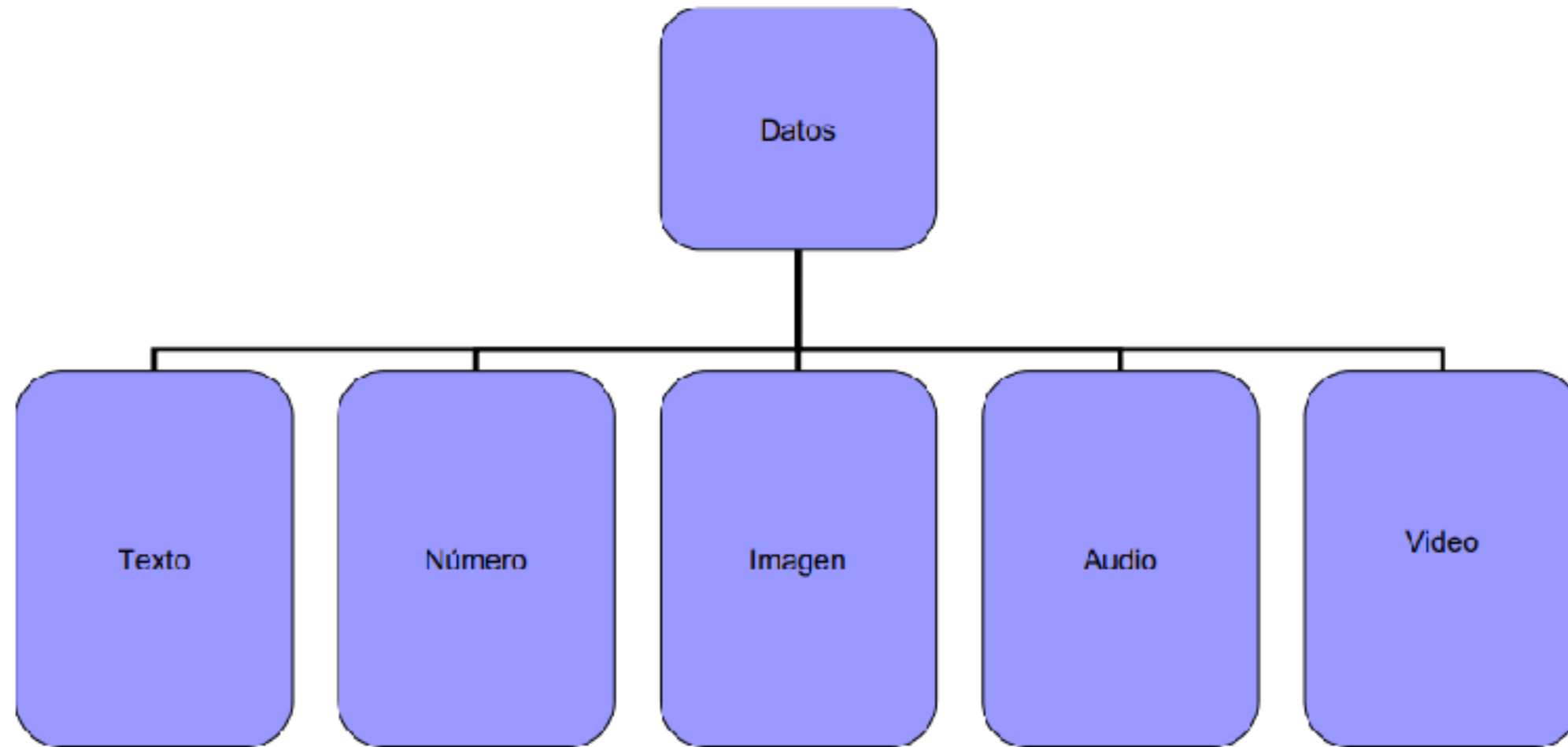


Introducción a la Programación y Computación 1 Sección E

Ing. MSc. Neftalí Calderón

Tipos de Datos



Representación de datos

Los datos por lo general son una mezcla de tipos.

Se utiliza una representación Uniforme de todos los tipos de datos.

Los datos de entrada a una computadora se transforman en la representación uniforme al ser utilizados y almacenados por la PC.

A esta representación uniforme o formato universal se le llama PATRÓN DE BITS.

```
char cadena[50]
```

Un solo bit no puede resolver el problema de la representación de datos, si cada pieza de datos (carácter) pudiera representarse por un 1 o un 0, entonces solo se necesitaría un bit.

Sin embargo se hace necesario almacenar números mas grandes, texto, gráficos y otros tipos de datos. Es aquí donde se hacen necesarios los PATRONES DE BITS.

Para representar diferentes tipos de datos se utiliza un PATRON DE BITS, una secuencia o como a veces se le llama, una cadena de bits.

Ejemplo: 1 0 0 0 1 0 1 0 1 1 1 1 1 1

Esto significa que si se quiere almacenar un patrón de bits formado por 16 bits, necesitan 16 interruptores electrónicos. Si quiere almacenar 1000 patrones de bits, cada uno de 16 bits, necesita 16 000 bits y así sucesivamente.

1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	
11	

0	0 0	0 0 0
1	0 1	0 0 1
	1 0	0 1 0
	1 1	0 1 1
		1 0 0
		1 0 1
		1 0 0
		1 1 1

Una pieza de texto en cualquier idioma es una secuencia de símbolos usados para representar una idea en ese idioma. (ej, ABC...Z, 0,1,2,3...9).

Se puede representar cada símbolo (de lenguajes humanos) con un patrón de bits (lenguaje de la maquina)

B Y T E

1000010 1011001 1010100 1000101

b y t e

0110010 1001001 1000100 0110101

La longitud del patrón de bits que representa un símbolo en un idioma depende del número de símbolos usados en ese idioma.

Códigos

Se han diseñado diferentes secuencias de patrones de bits para representar símbolos de texto. A esta secuencia se le conoce como Código y al proceso de representar los símbolos se le llama codificación.

ASCII

Código Americano de Estándares para Intercambio de Información (American Standard Code for Information Interchange).

Este código ASCII extendido utiliza 8 bits para cada símbolo, esto significa 256 símbolos distintos. Cada patrón cabe fácilmente en un byte de memoria

`'Hola' == 'hola'`

EBCDIC

Código Extendido de Intercambio decimal codificado en binario desarrollado por IBM al principio de la era de las computadoras. Utiliza patrones de ocho bits, de manera que puede representar hasta 256 símbolos. Solo es utilizado en computadoras Mainframes de IBM

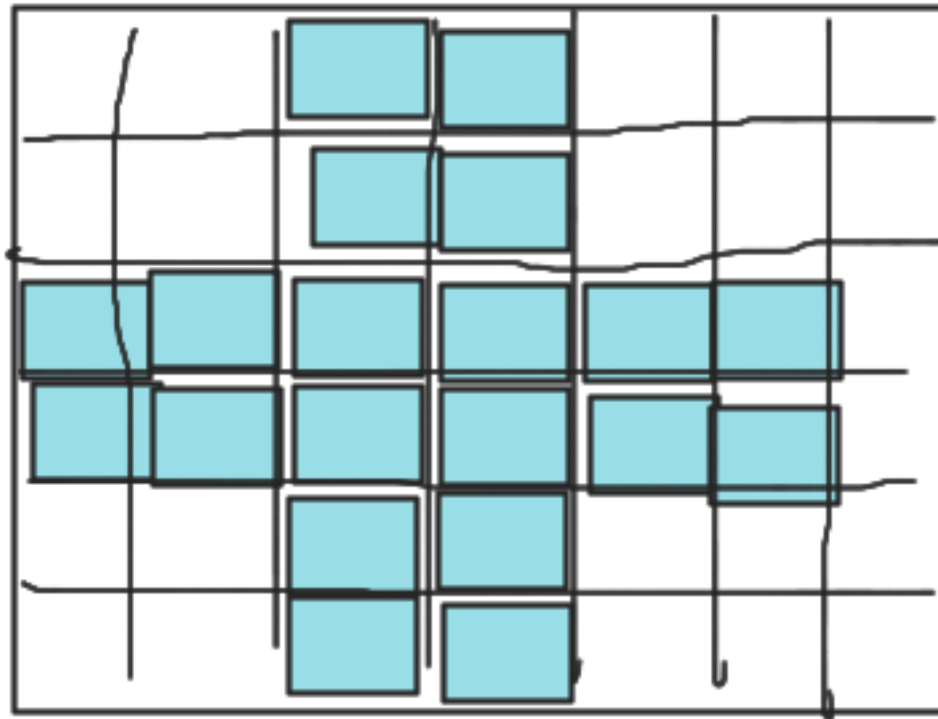
Unicode (UTF-8)

Ante la necesidad de un código de mayores capacidades, una coalición de fabricantes de hardware y software desarrollo un código que utiliza 16 bits y puede representar hasta 65 536 (2^{16}) símbolos. Diferentes secciones del código se asignan a los símbolos de distintos idiomas en el mundo.

ISO

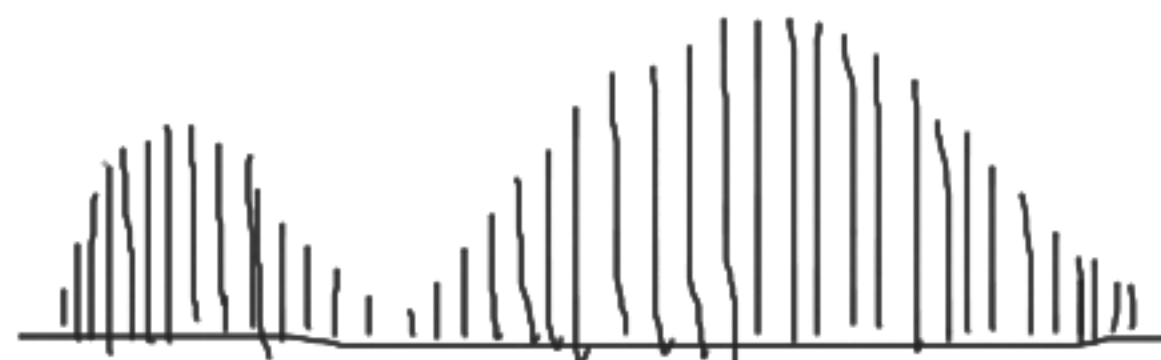
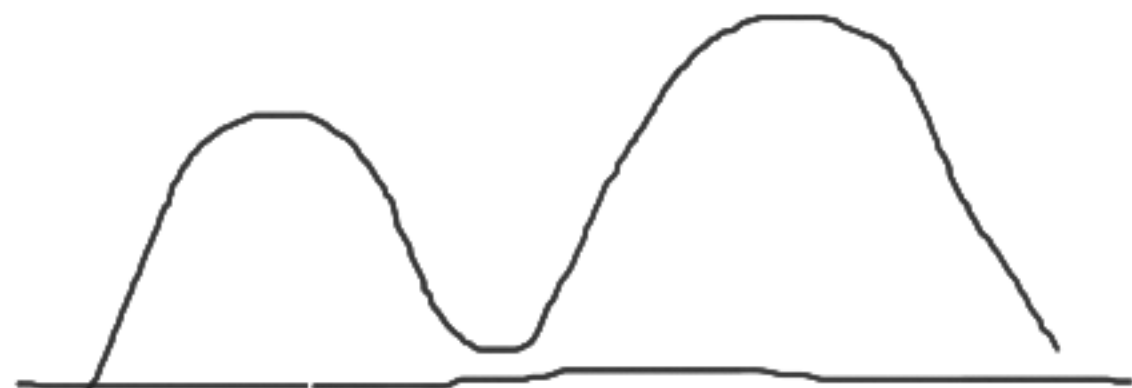
La Organización Internacional para la Estandarización ha diseñado un código que utiliza patrones de 32 bits . Este código representa hasta 4 294 967 296 símbolos, definitivamente lo suficiente para representar cualquier símbolo en el mundo actual.

Mapa de bits
Vectores



```
0 0 1 1 0 0 0
0 0 1 1 0 0 0
1 1 1 1 1 1 0
1 1 1 1 1 1 0
0 0 1 1 0 0 0
0 0 1 1 0 0 0
```

```
0 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0
```



0010101 1000110 0001010 1010100 0000000

clase

funcion1()

funcion2()

procedimiento1()

main()

```
int miFuncion(int a, b) {  
    int resultado;  
  
    resultado = a * b - 10 + 15;  
  
    return(resultado);  
}
```

```
leer(a);  
leer(b);
```

```
imprimir(a * b - 10 +15);
```

```
void miProcedimiento() {  
    leer(a);  
    leer(b);  
}
```

```
void main() {  
    int a;  
    int b;  
    int resultado;
```

```
    leer(a);  
    leer(b);
```

```
    resultado = miFuncion(a, b);  
    escribir(resultado);  
}
```

interacciones	1	2	3	4	5	6..... n
serie	2	4	8	16	32	64..... n
	2+2	4+4	8+8	16+16		

$n = 4$
 $\text{sig} = 2, 4, 8, 16, 32$
 $i = 0, 1, 2, 3, 4$

```

void serieNumerica(int n) {
    int sig = 2;

    for (int i=0; i<n; i++) {
        imprimir(sig + ' ');
        sig = sig*2;
    }
}

```

2 4 8 16

interacciones	1	2	3	4	5	6	n
serie	8	9	11	14	18	23....	n
		+1	+2	+3	+4	+5	

n = 4

sig = 8, 9, 11, 14, 18

i = 1, 2, 3, 4, 5

```
void serieNumerica(int n) {
    int sig = 8;

    for (int i=1; i<=n; i++) {
        imprimir(sig + ' ');
        sig = sig + i;
    }
}
```

8 9 11 14