

# **Fundamentos de Ingeniería Informática**

## **Módulo III - Unidad 9 Representación de la información**

Profesor: Héctor molina García

Versión: 0.1



Universidad  
Francisco de Vitoria  
UFV Madrid

*Grado en Ingeniería Informática  
Escuela Politécnica Superior*

¿Qué son los datos?

¿Qué es la información?

¿Es lo mismo datos e información?

# Información vs Datos

Datos



vs

Información



**Los datos** son elementos crudos  
desorganizados y sin refinar.

**La información** es la organización e  
interpretación de esos elementos.

# Información vs Datos

Ambos conceptos tienen un papel importante en Ciencias de la Computación, pero existen diferencias significativas entre ellos.

| Datos                                                                          | Información                                                                                     |
|--------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Los datos se refieren a hechos brutos que no tienen un significado específico. | La información se refiere a los datos procesados que tienen un propósito y significado.         |
| Los datos son independientes de la información.                                | La información depende de los datos.                                                            |
| Los datos o datos sin procesar no son suficientes para tomar una decisión.     | La información suele ser suficiente para ayudar a tomar una decisión en un contexto específico. |

# Información en ordenadores

# La información en los ordenadores

Un **bit** o **dígito binario** y es la unidad más pequeña de datos en una computadora. Los bits sólo pueden contener uno de dos valores: 0 o 1.

Un byte es una unidad de información formada por una secuencia de Bits. Según el contexto, puede representar diferentes tipos de información:

- Texto
- Número
- Instrucción del programa.
- Pixel en una imagen o parte de una grabación de audio.

# La información en los ordenadores

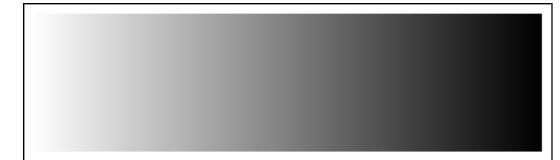


Imagen creada con 1 bit: 2 colores.

# La información en los ordenadores



La información se modela utilizando una escala de grises donde cada píxel puede representar un tono de gris (0 - 256).



Negro 00000000

Blanco 11111111

# La información en los ordenadores



La información se modela utilizando varias capas de color (rojo, verde, azul) y una para el factor de luminosidad.



Negro 00000000

Blanco 11111111

16.7 Millones de colores

# La información en los ordenadores



La información se modela utilizando varias capas de color (rojo, verde, azul) y una para el factor de luminosidad.



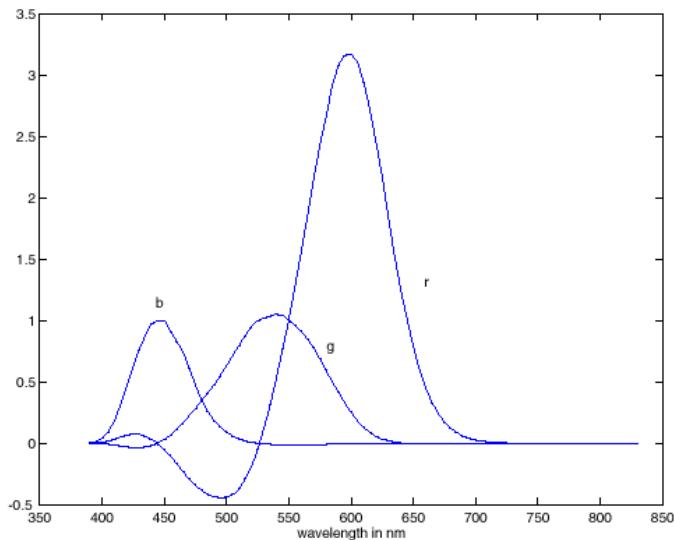
Negro 00000000

Blanco 11111111

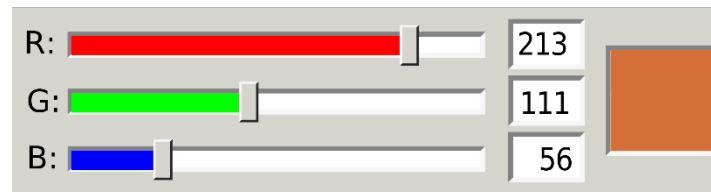
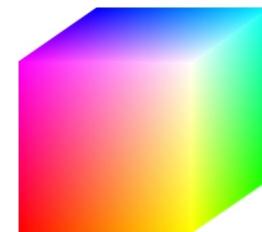
281 Trillón de colores

# La información en los ordenadores

El **modelo de color RGB** es un modelo de color aditivo en el que la luz roja, verde y azul se suman de varias maneras para reproducir una amplia gama de colores. El nombre del modelo proviene de las iniciales de los tres colores aditivos primarios / secundarios, rojo, verde y azul.



- $p_1 = 645.2 \text{ nm}$
- $p_2 = 525.3 \text{ nm}$
- $p_3 = 444.4 \text{ nm}$



Hex representation

- #FF0000
- #00FF00
- #0000FF
- #FFFF00
- #CCEEFF

Cada color está representado por un número hexadecimal de 2 dígitos.

# La información en los ordenadores

Un **sistema numérico** es un modo de escritura para expresar números; es decir, una notación matemática para representar números de un conjunto dado, usando dígitos u otros símbolos de manera consistente..

- Representar un conjunto útil de números.
- Darle a cada número representado una representación única (o al menos una representación estándar).
- Reflejar la estructura algebraica y aritmética de los números.

El **cardinal** (número de elementos) de un conjunto de números se llama la base de un sistema numérico..

Sistema decimal: base = 10, dígitos = {0,1,2,3,4,5,6,7,8,9}

# La información en los ordenadores

La representación posicional estándar de un número N en base b se escribe de la siguiente manera:

$$N = (a_n a_{n-1} a_{n-2} \dots a_1 a_0 a_{-1} \dots a_{-m})_b$$

Dónde:

- $a_i \rightarrow$  dígitos que constituyen el número (entre 0 y r-1)
- $n \rightarrow$  Número de dígitos enteros
- $m \rightarrow$  Número de dígitos fraccionarios
- $a_n \rightarrow$  Dígito más significativo
- $a_{-m} \rightarrow$  dígito menos significativo
- $r^i \rightarrow$  Peso del dígitoi
- $a_i * r^i \rightarrow$  Valor del dígitoi

# La información en los ordenadores

El valor de un número N viene dado por:

$$(N)_b = a_{n-1} \times b^{n-1} + a_{n-2} \times b^{n-2} \dots + a_1 \times b^1 + a_0 \times b^0 + a_{-1} \times b^{-1} \dots + a_{-m} \times b^{-m}$$

donde "b" es la base del sistema numérico (por ejemplo, 2, 8, 10 o 16) y "a" es un dígito que va de 0 a b-1.

$$\begin{aligned}(352.45)_{10} &= 3 \times 10^2 + 5 \times 10^1 + 2 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} \\ &= 3 \times 100 + 5 \times 10 + 2 \times 0 + 4 \times 0,1 + 5 \times 0,01\end{aligned}$$

# La información en los ordenadores

**La conversión base** es el proceso de convertir un número N de un número base a otro número base. Basta con expresar el número a convertir en notación polinómica, expresando los dígitos y pesos en base s, y operar en base s:

$$N = (10101)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (21)_{10}$$

$$M = (14)_{16} = 1 \times 16^1 + 4 \times 16^0 = (20)_{10}$$

# De binario a decimal

# De binario a decimal

## Sistemas posicionales (numerales)

- Binario

- Decimal

- Hexadecimal

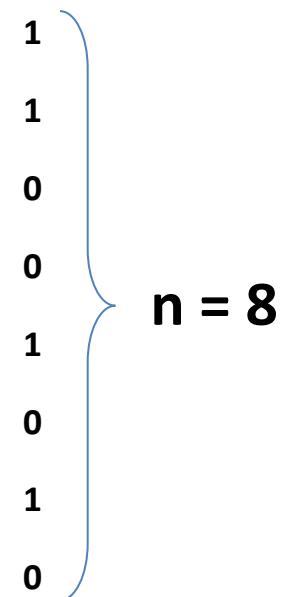
¿Cómo podemos numerar n<sub>2</sub> convertir de una base 2 a  
base 10?

$$n_2 = 11001010$$

# De binario a decimal

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Hexadecimal



A binary number consisting of eight digits: 1, 1, 0, 0, 1, 0, 1, 0. A blue brace on the right side groups all eight digits together, and the label  $n = 8$  is placed next to the brace.

Primero contamos el número de dígitos en nuestro número binario.

# De binario a decimal

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Hexadecimal

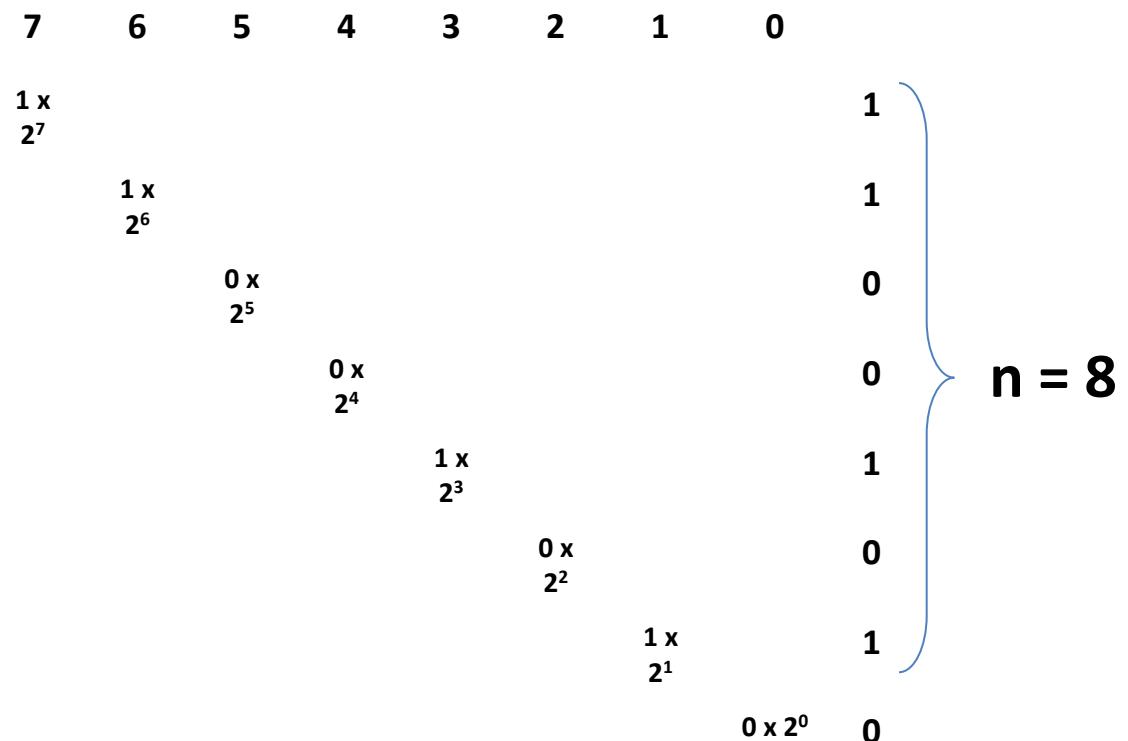
A binary number consisting of eight digits: 1, 1, 0, 0, 1, 0, 1, 0. A blue brace on the right side groups the last seven digits (0, 0, 1, 0, 1, 0, 1) and is labeled  $n = 8$ .

A continuación, multiplicamos de izquierda a derecha cada dígito por la potencia de dos que le corresponde, empezando por  $2^{n-1}$

## De binario a decimal

## Sistemas posicionales (numerales)

- Binario
  - Decimal
  - Hexadecimal

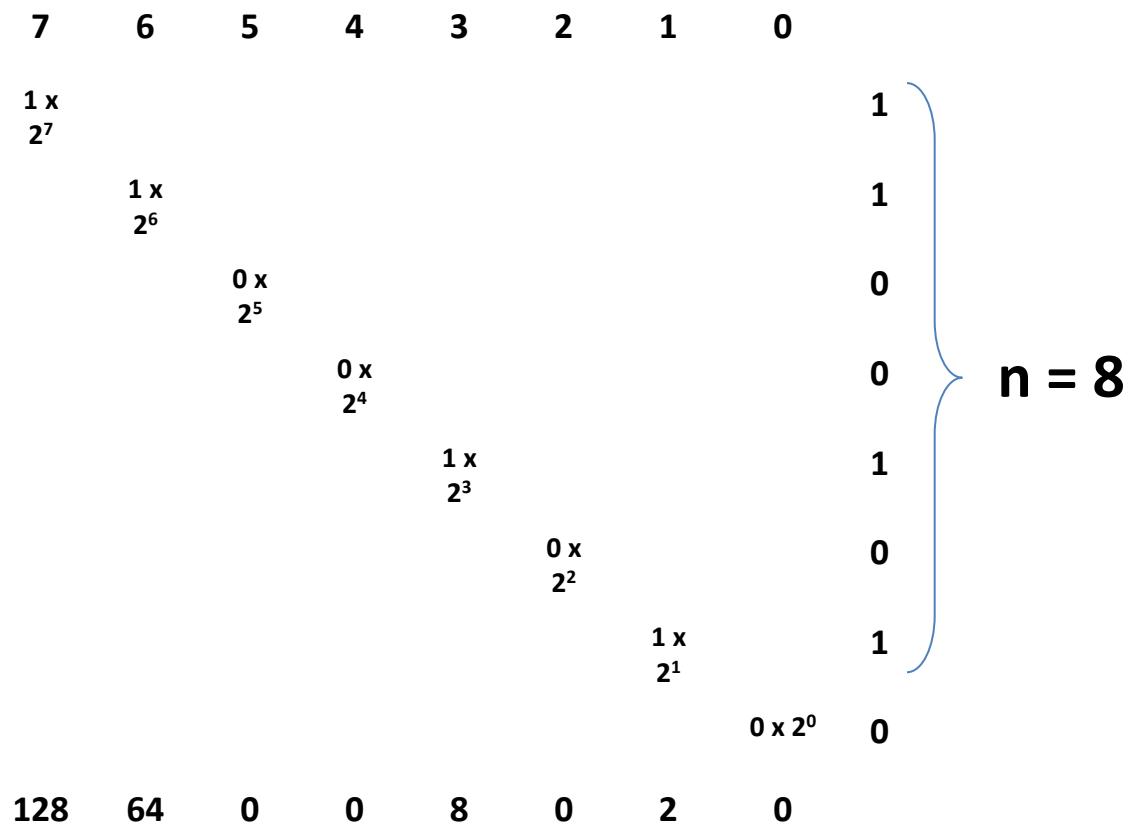


A continuación, multiplicamos de izquierda a derecha cada dígito por la potencia de dos que le corresponde, empezando por  $2^{n-1}$

## De binario a decimal

# Sistemas posicionales (numerales)

- Binario
  - Decimal
  - Hexadecimal

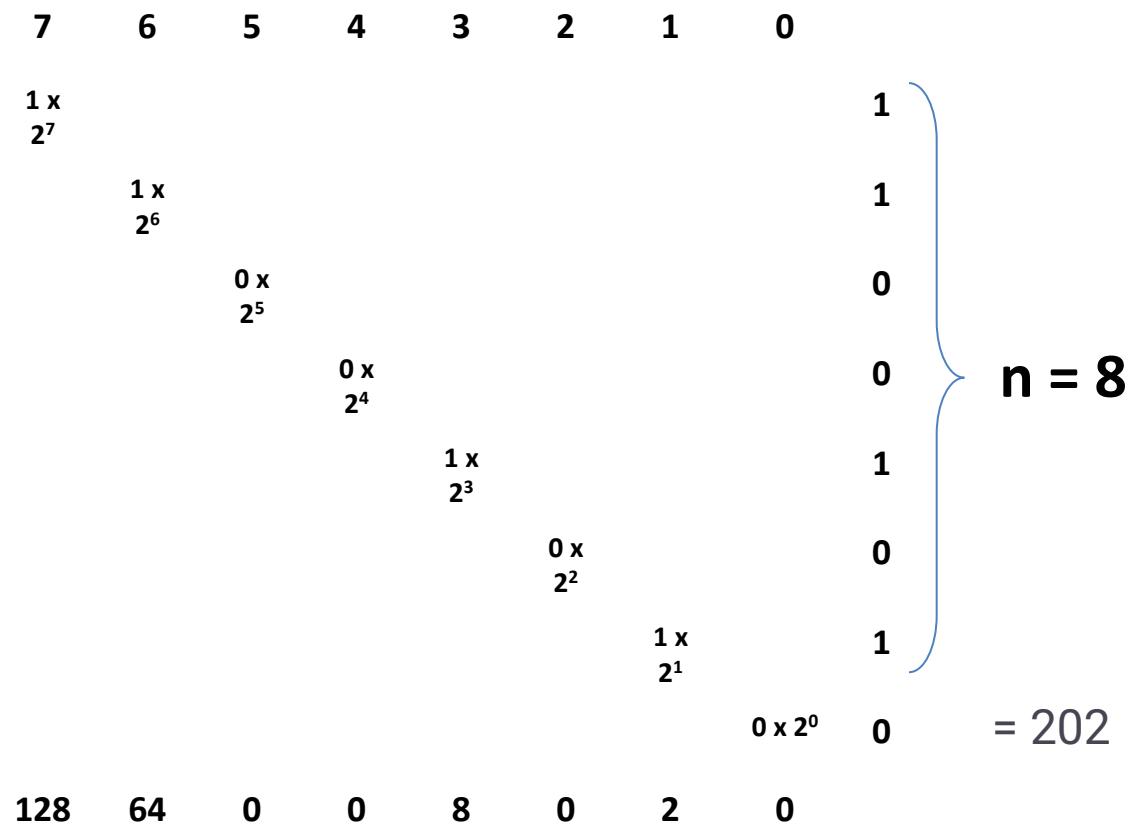


Finalmente, añadimos todos esos valores.

# De binario a decimal

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Hexadecimal



# De binario a Octal

# De binario a octal

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Octal

¿Cómo podemos numerar  $n_2$  convertir de una base 2 a  
base 8?

$$n_2 = 100111011101111$$

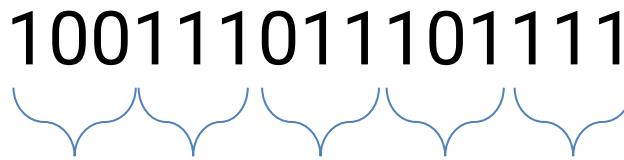
# De binario a octal

Sistemas  
(numerales)

- Binario
- Decimal
- Octal

posicionales

100111011101111



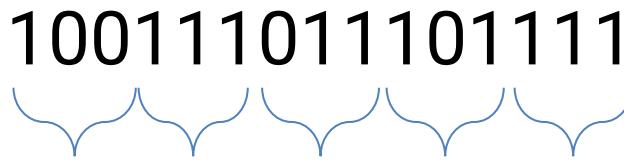
Dividimos los bits en grupos de 3 de derecha a izquierda. ¿Por qué?

# De binario a octal

Sistemas  
(numerales)

- Binario
- Decimal
- Octal

posicionales

100111011101111  


Dividimos los bits en grupos de 3 de derecha a izquierda.  
¿Por qué?

**Necesitamos 3 bits para representar números del 0  
al 7.**

# De binario a octal

Sistemas  
(numerales)

- Binario
- Decimal
- Octal

posicionales

100111011101111

A binary number 100111011101111 is displayed. Blue curly braces are placed under the number to group it into three-digit segments: (100) (111) (011) (110) (111). This visual cue indicates that each group of three binary digits will be converted into a single octal digit.

Convertimos cada triplete a su equivalente octal de un solo dígito.

# De binario a octal

Sistemas  
(numerales)

- Binario
- Decimal
- Octal

posicionales

100111011101111



| Dec | Hex | Oct | Bin  |
|-----|-----|-----|------|
| 0   | 0   | 000 | 0000 |
| 1   | 1   | 001 | 0001 |
| 2   | 2   | 002 | 0010 |
| 3   | 3   | 003 | 0011 |
| 4   | 4   | 004 | 0100 |
| 5   | 5   | 005 | 0101 |
| 6   | 6   | 006 | 0110 |
| 7   | 7   | 007 | 0111 |
| 8   | 8   | 010 | 1000 |
| 9   | 9   | 011 | 1001 |
| 10  | A   | 012 | 1010 |
| 11  | B   | 013 | 1011 |
| 12  | C   | 014 | 1100 |
| 13  | D   | 015 | 1101 |
| 14  | E   | 016 | 1110 |
| 15  | F   | 017 | 1111 |

Convertimos cada triplete a su equivalente octal de un solo dígito.

# De binario a octal

Sistemas  
(numerales)

- Binario
- Decimal
- Octal

posicionales

100111011101111  
7

| Dec | Hex | Oct | Bin  |
|-----|-----|-----|------|
| 0   | 0   | 000 | 0000 |
| 1   | 1   | 001 | 0001 |
| 2   | 2   | 002 | 0010 |
| 3   | 3   | 003 | 0011 |
| 4   | 4   | 004 | 0100 |
| 5   | 5   | 005 | 0101 |
| 6   | 6   | 006 | 0110 |
| 7   | 7   | 007 | 0111 |
| 8   | 8   | 010 | 1000 |
| 9   | 9   | 011 | 1001 |
| 10  | A   | 012 | 1010 |
| 11  | B   | 013 | 1011 |
| 12  | C   | 014 | 1100 |
| 13  | D   | 015 | 1101 |
| 14  | E   | 016 | 1110 |
| 15  | F   | 017 | 1111 |

# De binario a octal

Sistemas  
(numerales)

- Binario
- Decimal
- Octal

posicionales

100111011101111  
4      7      3      5      7

| Dec | Hex | Oct | Bin  |
|-----|-----|-----|------|
| 0   | 0   | 000 | 0000 |
| 1   | 1   | 001 | 0001 |
| 2   | 2   | 002 | 0010 |
| 3   | 3   | 003 | 0011 |
| 4   | 4   | 004 | 0100 |
| 5   | 5   | 005 | 0101 |
| 6   | 6   | 006 | 0110 |
| 7   | 7   | 007 | 0111 |
| 8   | 8   | 010 | 1000 |
| 9   | 9   | 011 | 1001 |
| 10  | A   | 012 | 1010 |
| 11  | B   | 013 | 1011 |
| 12  | C   | 014 | 1100 |
| 13  | D   | 015 | 1101 |
| 14  | E   | 016 | 1110 |
| 15  | F   | 017 | 1111 |

# De binario a hexadecimal

# De binario a hexadecimal

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Hexadecimal

¿Cómo podemos numerar  $n_2$  convertir de una base 2 a  
base 16?

$$n_2 = 100111011101111$$

# De binario a hexadecimal

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Hexadecimal

100111011101111

Dividimos los bits en grupos de 4. ¿Por qué?

# De binario a hexadecimal

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Hexadecimal

100111011101111



| Dec | Hex | Oct | Bin  |
|-----|-----|-----|------|
| 0   | 0   | 000 | 0000 |
| 1   | 1   | 001 | 0001 |
| 2   | 2   | 002 | 0010 |
| 3   | 3   | 003 | 0011 |
| 4   | 4   | 004 | 0100 |
| 5   | 5   | 005 | 0101 |
| 6   | 6   | 006 | 0110 |
| 7   | 7   | 007 | 0111 |
| 8   | 8   | 010 | 1000 |
| 9   | 9   | 011 | 1001 |
| 10  | A   | 012 | 1010 |
| 11  | B   | 013 | 1011 |
| 12  | C   | 014 | 1100 |
| 13  | D   | 015 | 1101 |
| 14  | E   | 016 | 1110 |
| 15  | F   | 017 | 1111 |

Dividimos los bits en grupos de 4. ¿Por qué?

Necesitamos 4 bits en binario para representar 16 valores en hexadecimal.

# De binario a hexadecimal

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Hexadecimal

100111011101111  
4 E E F

| Dec | Hex | Oct | Bin  |
|-----|-----|-----|------|
| 0   | 0   | 000 | 0000 |
| 1   | 1   | 001 | 0001 |
| 2   | 2   | 002 | 0010 |
| 3   | 3   | 003 | 0011 |
| 4   | 4   | 004 | 0100 |
| 5   | 5   | 005 | 0101 |
| 6   | 6   | 006 | 0110 |
| 7   | 7   | 007 | 0111 |
| 8   | 8   | 010 | 1000 |
| 9   | 9   | 011 | 1001 |
| 10  | A   | 012 | 1010 |
| 11  | B   | 013 | 1011 |
| 12  | C   | 014 | 1100 |
| 13  | D   | 015 | 1101 |
| 14  | E   | 016 | 1110 |
| 15  | F   | 017 | 1111 |

Convertimos cada grupo de cuatro bits a un hexadecimal.

# De decimal a binario

# De decimal a binario

## Sistemas posicionales (numerales)

- Binario
- Decimal

¿Cómo podemos numerar  $n_1$  convertir de una base 10 a base 2?

$$n_1 = 233$$

- Hexadecimal

# De decimal a binario

## Sistemas posicionales (numerales)

- Binario
- Decimal
- Hexadecimal

233

base

/  2

# De decimal a binario

## Sistemas posicionales (numerales)

- Binario

233  
1      116  
remainder

base

/ 2

- Decimal

- Hexadecimal

Dividimos por la base que queremos transformar y almacenamos el resto.

# De decimal a binario

## Sistemas posicionales (numerales)

- Binario

|       |     |
|-------|-----|
| 233   | / 2 |
| 1 116 | / 2 |
| 0 58  | / 2 |
| 0 29  | / 2 |
| 1 14  | / 2 |
| 0 7   | / 2 |
| 1 3   | / 2 |
| 1 1   | / 2 |
| 1     |     |

- Decimal

- Hexadecimal

# De decimal a binario

## Sistemas posicionales (numerales)

- Binario

|                 |     |
|-----------------|-----|
| 233             | / 2 |
| 1 116           | / 2 |
| 0 58            | / 2 |
| 0 29            | / 2 |
| 1 14            | / 2 |
| 0 7             | / 2 |
| 1 3             | / 2 |
| 1 1             | / 2 |
| 1               |     |
| 1 0 0 1 0 1 1 1 |     |

- Decimal

- Hexadecimal

¿10010111 es 233 en base decimal?

10010111 es 233 en base decimal?

NO, 10010111 es 151.

# De decimal a binario

## Sistemas posicionales (numerales)

- Binary

|       |     |
|-------|-----|
| 233   | / 2 |
| 1 116 | / 2 |
| 0 58  | / 2 |
| 0 29  | / 2 |
| 1 14  | / 2 |
| 0 7   | / 2 |
| 1 3   | / 2 |
| 1 1   | / 2 |
| 1     |     |

- Decimal

- Hexadecimal

Tenemos que revertir el resto.

# Sistemas posicionales (numerales)

Algunas preguntas importantes ....

¿Cuántos valores pueden ser representados por n bits?

# Sistemas posicionales (numerales)

Algunas preguntas importantes ....

¿Cuántos valores pueden ser representados por n bits?  $2^n$

# Sistemas posicionales (numerales)

Algunas preguntas importantes ....

¿Cuántos valores pueden ser representados por n bits?

$$2^n$$

¿Cuántos bits se necesitan para representar m valores?

# Sistemas posicionales (numerales)

Algunas preguntas importantes ....

¿Cuántos valores pueden ser representados por n bits?  $2^n$

¿Cuántos bits se necesitan para representar m valores?

$$\begin{aligned} & \text{Log}_2(n) \text{ por exceso} \\ & \text{Log}_2(91) = 6.50779 = 7 \end{aligned}$$

# Sistemas posicionales (numerales)

Algunas preguntas importantes ....

¿Cuántos valores pueden ser representados por n bits?  $2^n$

¿Cuántos bits se necesitan para representar m valores?

$$\begin{aligned} & \text{Log}_2(n) \text{ por exceso} \\ & \text{Log}_2(91) = 6.50779 = 7 \end{aligned}$$

Si usamos n bits, si corresponde el valor mínimo representable

Al número 0, ¿cuál es el valor numérico máximo representable?

# Sistemas posicionales (numerales)

Algunas preguntas importantes ....

¿Cuántos valores pueden ser representados por n bits?  $2^n$

¿Cuántos bits se necesitan para representar m valores?  
 $\log_2(n)$  por exceso  
 $\log_2(91) = 6.50779 = 7$

Si usamos n bits, si corresponde el valor mínimo representable  
Al número 0, ¿cuál es el valor numérico máximo representable?

$$2^n - 1$$

# Sistemas posicionales (numerales)

No se puede usar ninguna calculadora en el examen.

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$



# Sistemas posicionales (numerales)

| 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           |



# Representación numérica (enteros)

## Representación numérica (enteros)

Los números enteros se representan mediante un número fijo de bits. El rango de números representables depende del ancho y la convención de representación (es decir, no todos los números pueden ser representados, sólo aquellos que están dentro del rango).

Rango representable de números naturales (enteros sin signo): $[0, 2^n - 1]$ , donde n es el ancho.

Ejemplo: Si usamos un ancho  $n = 4$ , se pueden representar los números naturales en el rango  $[0, 15]$ .

# Representación numérica (enteros)

## Convención de signos y magnitudes

El sistema decimal define el signo añadiendo un símbolo a la magnitud para representar el signo del número.

En binario, el signo está representado por un bit: 0 (+), 1 (-)

$(+16)_{10}$  = 00010000 en binario con 8 bits

$(-16)_{10}$  = 10010000 en binario con 8 bits

Rango de números representables con n bits:  $[-2^{n-1}, 2^{n-1}]$ .

**Si usamos 8 bits con signo, podemos representar números enteros en el rango [-127, +127].**

# Representación numérica (enteros)

## Convención de signos y magnitudes

Para calcular el valor entero de un número con signo binario, debemos seguir los siguientes pasos:

1. Convertir la magnitud a base 10 usando los bits menos significativos  $n-1$ .
2. Agregue el signo: + (si comienza con 0) o - (si comienza con 1).

Ejemplo:  $00111 = 7$ ,  $11010 = -10$

- Las operaciones de suma y resta son más complicadas en binario, ya que los signos y las magnitudes deben tenerse en cuenta por separado.
- El cero tiene dos representaciones  $[-0, +0]$ .

# Representación numérica (enteros)

## Convención de signos y magnitudes

- Los números positivos se representan como una magnitud y un signo (por lo tanto, comenzando con 0).
- Los números negativos se representan como el complemento del 2 del número positivo correspondiente (comenzando con 1).

**Rango de números representables usando n bits:  $[-2^{n-1}, 2^{n-1}-1]$**

**Si usamos 8 bits, podemos representar números enteros en el rango  $[-127, +127]$ .**

La representación posicional estándar de un número N en base b usando el complemento de dos se escribe de la siguiente manera::

$$N = (a_{n-1}a_{n-2} \dots a_1a_0 a_{-1} \dots a_{-m})_b$$

# Representación numérica (enteros)

## COMPLEMENTO A DOS

**El complemento a dos** es una operación matemática para convertir reversiblemente un número binario positivo en un número binario negativo con valor equivalente (pero negativo), utilizando el dígito binario con el mayor valor posicional para indicar si el número binario es positivo o negativo (el signo).

$$C_b(N) = b^n - N$$

El total de números positivos será  $2^{n-1} - 1$  y el total de negativos será  $2^{n-1}$  donde n es el número máximo de bits. El 0 contaría por separado.

Si usamos 4 dígitos →  $C_{10}(0129) = 10^4 - 0129 = 9871$

$$9871 + 0129 = 10000 = 10^4$$

# Representación numérica (enteros)

## De binario al complemento a dos

**El complemento a dos** es una operación matemática para convertir reversiblemente un número binario positivo en un número binario negativo con valor equivalente (pero negativo), utilizando el dígito binario con el mayor valor posicional para indicar si el número binario es positivo o negativo (el signo).

1. Quita el signo y usa el número positivo
2. Convierte el número decimal en un número binario.
3. Traspón todos los dígitos a partir de la parte significativa: ceros en unos y unos en ceros.
4. Añade 1 si el número es negativo.

# Representación numérica (enteros)

## Convertir de binario a complemento a dos

| Decimal signed number | Positive binary | Negative binary |
|-----------------------|-----------------|-----------------|
| 0                     | 0000            | 0000            |
| 1                     | 0001            | 1110            |
| 2                     | 0010            | 1101            |
| 3                     | 0011            | 1100            |
| 4                     | 0100            | 1011            |
| 5                     | 0101            | 1010            |
| 6                     | 0110            | 1001            |
| 7                     | 0111            | 1000            |
| 8                     |                 | 1111            |

**Si usamos 4 bits, podemos representar números enteros en el rango [-8, +7].**

# Representación numérica (enteros)

Convertir -68 a binario en el complemento de dos

**¿Cuántos bits necesito para representar -68 en binario?**

# Representación numérica (enteros)

Convertir -68 a binario en el complemento de dos

**¿Cuántos bits necesito para representar -68 en binario?**

**8 bits**

Necesitamos 7 bits para representar 68 pero necesito otro poco más para representar -68 para obtener un rango entre [-128 + 127].

# Representación numérica (enteros)

Convertir -68 a binario en el complemento de dos

¿Cuántos bits necesito para representar -68 en binario?

8 bits

Necesitamos 7 bits para representar 68 pero necesito otro poco más para representar -68 para obtener un rango entre [-128 + 127].

|    |    |    |   |   |   |   |  |   |   |
|----|----|----|---|---|---|---|--|---|---|
| 68 |    |    |   |   |   |   |  | / | 2 |
| 0  | 34 |    |   |   |   |   |  | / | 2 |
|    | 0  | 17 |   |   |   |   |  | / | 2 |
|    |    | 1  | 8 |   |   |   |  | / | 2 |
|    |    |    | 0 | 4 |   |   |  | / | 2 |
|    |    |    |   | 0 | 2 |   |  | / | 2 |
|    |    |    |   |   | 0 | 1 |  | / | 2 |
|    |    |    |   |   |   |   |  | / | 2 |
| 0  | 0  | 1  | 0 | 0 | 0 | 1 |  | / | 2 |

$$68 = 01000100$$

Agrega un 0 extra para tener 8 bits.

# Representación numérica (enteros)

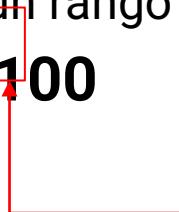
Convertir -68 a binario en el complemento de dos

¿Cuántos bits necesito para representar -68 en binario?

8 bits

Necesito 7 bits para representar 68 pero necesito otro poco más para representar -68 para obtener un rango entre [-128 + 127].

$68 = 01000100$



Encontramos la parte más significativa.

El primer 1 comenzando a la derecha.

$68 = 01000100$

Volteamos todos los bits después del bit más significativo.

$-68 = 10111100$

# Representación numérica (flotante)

# Representación numérica (flotante)

Los números reales se representan en computadoras aproximadamente usando el estándar IEEE-754, usando un entero con una precisión fija, llamado mantisa, escalado por un exponente entero de una base fija (notación científica).

Ejemplo: Si queremos representar 14.345 como número de coma flotante en base 10:

$$14.345 = \underbrace{14345}_{\text{mantisa}} \times \underbrace{10^{-3}}_{\text{base}}$$

exponente

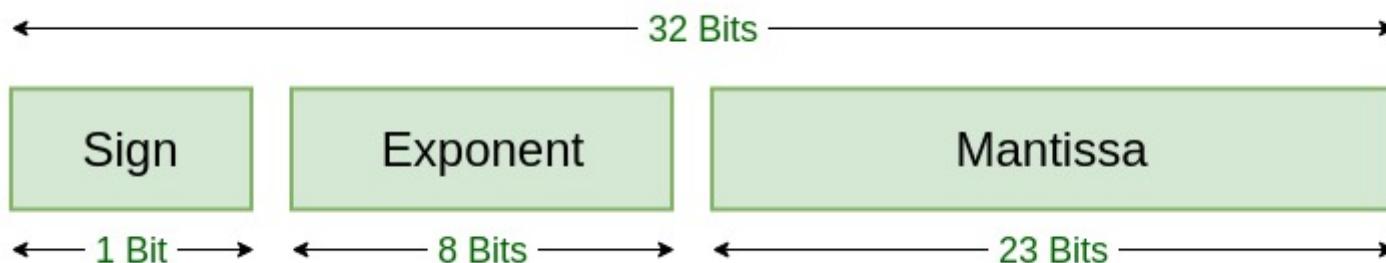
# Representación numérica (flotante)

Los números de coma flotante están representados en las computadoras por una colección finita de bits compuesta de tres partes:

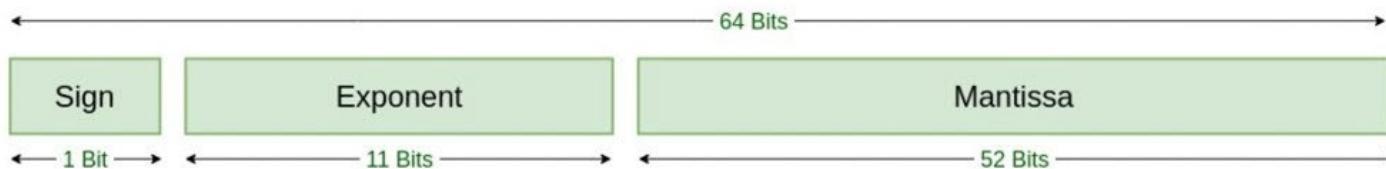
- Signo (1 bit simple): El bit de signo es 1 si un número es negativo y 0 si el número es positivo, como los enteros.
- Mantissa o significante o fracción (23 bits en coma flotante de precisión simple): La mantissa son los dígitos significativos del número en la representación de punto flotante.
- Exponente (coma flotante de precisión simple de 8 bits): El exponente es el radio que se eleva al determinar el valor de esa representación de punto flotante.

# Representación numérica (flotante)

Los números flotantes se dividen en dos basados en los tres componentes anteriores: precisión simple (32 bits) y precisión doble (64 bits).



Precisión simple



Doble precisión

# Representación numérica (flotante)

Para convertir números decimales en representación de coma flotante IEEE 754:

1. Elija la representación de precisión: simple o doble.
2. Separar la parte entera y la parte decimal del número.
3. Convertir el número decimal en binario.
4. Convertir la parte decimal en binario.
5. Combinar las dos partes del número que se han convertido en binario.
6. Identificar el signo: 0 para números positivos y 1 para números negativos.

# Representación numérica (flotante)

Para convertir números decimales en representación de coma flotante IEEE 754:

7. Convertir el número binario en notación científica en base 2.
  - Para convertir el número en notación científica en base 2, debemos mover el punto decimal hacia la izquierda hasta que esté a la derecha del primer bit para crear la mantisa normalizada..
8. Calcular el exponente basándose en la precisión.

# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

**¿Qué precisión debemos usar?**

**Individual (32 bits)**

# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?



# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

88.125 (32 bits)  
Parte decimal

Número decimal

|                                             |   |   |
|---------------------------------------------|---|---|
| 88                                          | / | 2 |
| 0      44                                   | / | 2 |
| 0      22                                   | / | 2 |
| 0      11                                   | / | 2 |
| 1      5                                    | / | 2 |
| 1      2                                    | / | 2 |
| 0      1                                    | / | 2 |
|                                             | / | 2 |
| 0      0      0      1      1      0      1 |   |   |

# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

 88.125 (32 bits)

Número decimal

|                                             |   |   |
|---------------------------------------------|---|---|
| 88                                          | / | 2 |
| 0      44                                   | / | 2 |
| 0      22                                   | / | 2 |
| 0      11                                   | / | 2 |
| 1      5                                    | / | 2 |
| 1      2                                    | / | 2 |
| 0      1                                    | / | 2 |
|                                             | / | 2 |
| 0      0      0      1      1      0      1 |   |   |

$$88 = 1011000$$

# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?



|       |      |   |   |
|-------|------|---|---|
| 0.125 |      | * | 2 |
| 0     | 0.25 | * | 2 |
| 0     | 0.5  | * | 2 |
| 1     | 1.0  | * | 2 |

$$0.125 = 001$$

# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)

88.125

1011000.001 x 2<sup>0</sup>

Mueve el decimal 6 lugares a la izquierda para dejar un 1.

1.011000001 x 2<sup>0+6</sup>

Hay sesgos establecidos para la precisión simple y doble. El sesgo de exponente para la precisión simple es 127, lo que significa que debemos agregarle el exponente de base 2 que se encontró anteriormente. Por lo tanto, el exponente que usará es 127 + 6 que es 133.

127 + 6 = 133

# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)

$$127 + 6 = 133$$

|     |    |    |   |   |
|-----|----|----|---|---|
| 133 |    |    | / | 2 |
| 1   | 66 |    | / | 2 |
| 0   | 33 |    | / | 2 |
|     | 1  | 16 | / | 2 |
|     | 0  | 8  | / | 2 |
|     | 0  | 4  | / | 2 |
|     | 0  | 2  | / | 2 |
|     | 0  | 1  | / | 2 |
| 1   | 0  | 1  | 0 | 0 |
| 1   | 0  | 0  | 0 | 1 |

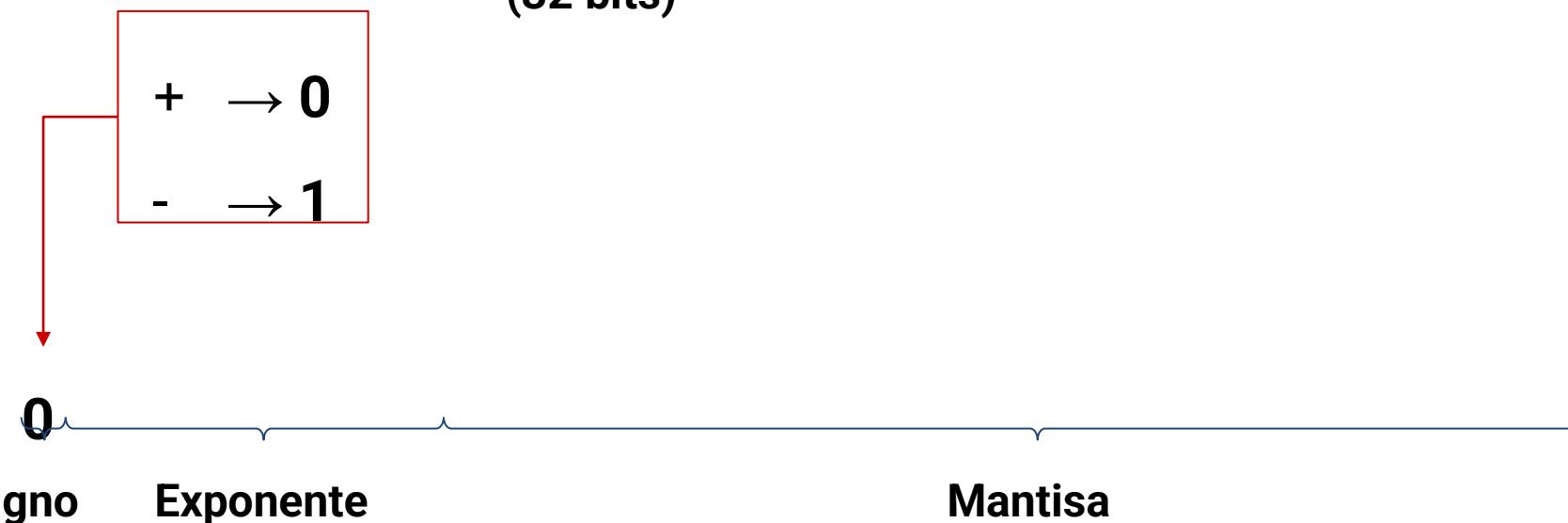
$$133 = 10000101$$

# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)



# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)

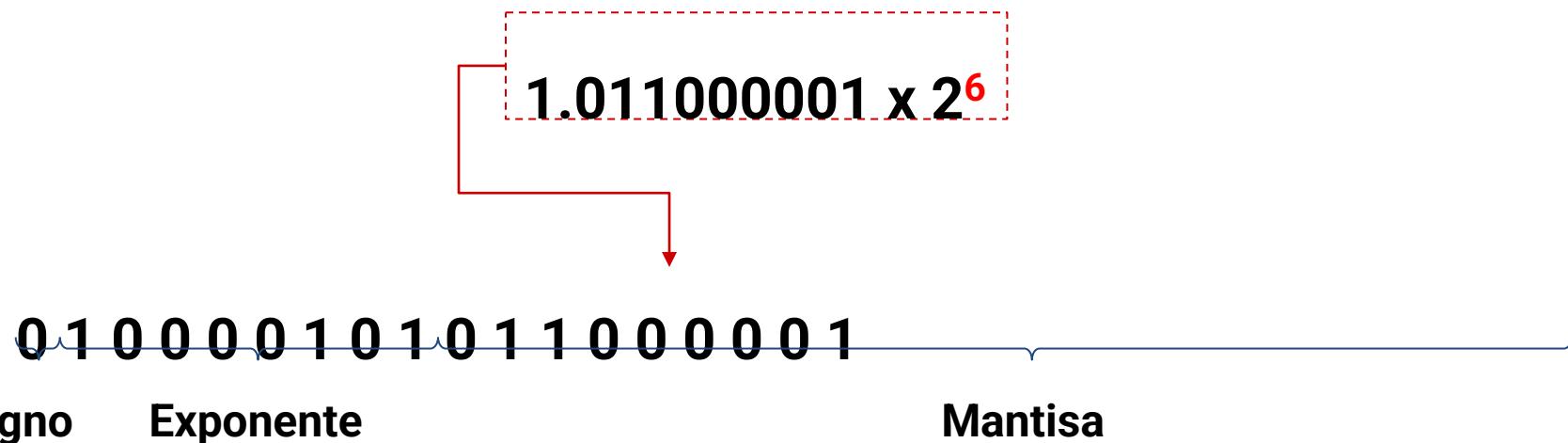


# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)



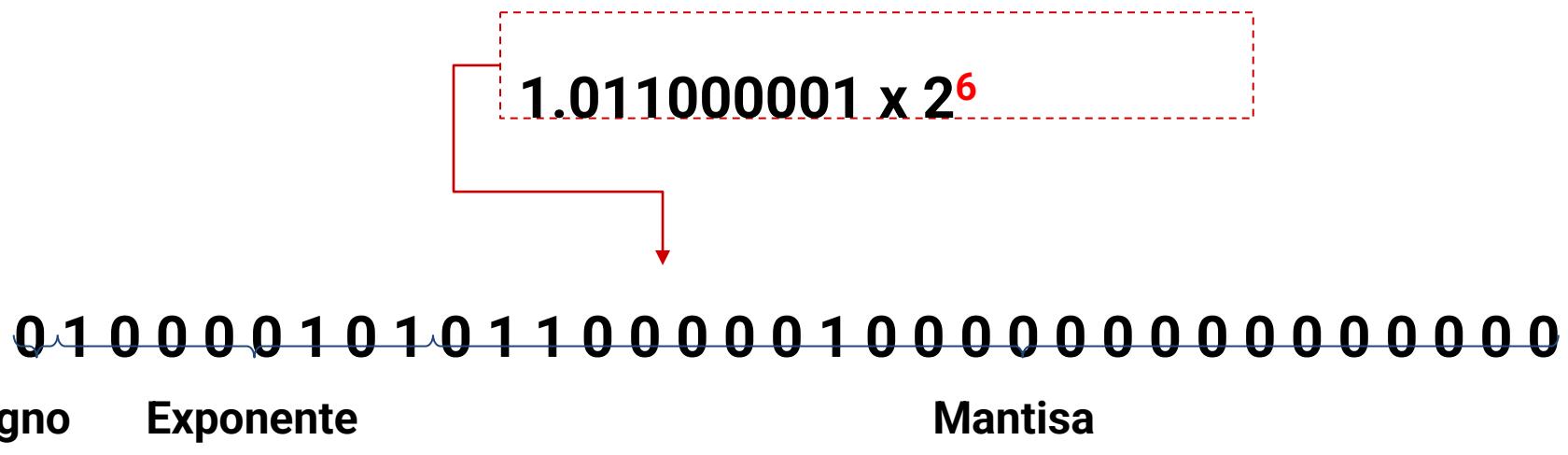
Simplemente soltamos el 1 a la izquierda y copiamos la parte decimal del número que se está multiplicando por 2.

# Representación numérica (flotante)

Convertir 88.125 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)



Completamos con ceros.

# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

**¿Qué precisión debemos usar?**

**(32 bits)**

# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?



# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?



|    |    |   |   |   |   |   |
|----|----|---|---|---|---|---|
| 25 |    |   |   |   | / | 2 |
| 1  | 12 |   |   |   | / | 2 |
| 0  | 6  |   |   |   | / | 2 |
|    | 0  | 3 |   |   | / | 2 |
|    |    | 1 | 1 |   | / | 2 |
| 1  | 0  | 0 | 1 | 1 |   |   |

# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?



|    |    |   |   |   |   |   |
|----|----|---|---|---|---|---|
| 25 |    |   |   |   | / | 2 |
| 1  | 12 |   |   |   | / | 2 |
| 0  | 6  |   |   |   | / | 2 |
|    | 0  | 3 |   |   | / | 2 |
|    |    | 1 | 1 |   | / | 2 |
| 1  | 0  | 0 | 1 | 1 |   |   |

# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?



|       |   |   |
|-------|---|---|
| 0.025 | * | 2 |
| 0     | * | 2 |
| 0.05  | * | 2 |
| 0     | * | 2 |
| 0.1   | * | 2 |
| 0     | * | 2 |
| 0.2   | * | 2 |
| 0     | * | 2 |
| 0.4   | * | 2 |
| 1     | * | 2 |
| 0.6   | * | 2 |
| 1     | * | 2 |
| 0.2   | * | 2 |
| 0     | * | 2 |
| 0.4   | * | 2 |

# Representación numérica (flotante)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

¿Qué precisión debemos usar?



|     |     |   |   |
|-----|-----|---|---|
| 0.4 |     | * | 2 |
| 0   | 0.8 | * | 2 |
| 1   | 0.6 | * | 2 |
| 1   | 0.2 | * | 2 |
| 0   | 0.4 | * | 2 |
| 0   | 0.8 | * | 2 |
| 1   | 0.6 | * | 2 |
| 1   | 0.2 | * | 2 |
| 0   | 0.4 | * | 2 |

# Representación numérica (flotante)

Convert 25.025 to binary using single IEEE 754 Floating Point Representation

¿Qué precisión debemos usar?



|     |     |   |   |
|-----|-----|---|---|
| 0.4 |     | * | 2 |
| 0   | 0.8 | * | 2 |
| 1   | 0.6 | * | 2 |
| 1   | 0.2 | * | 2 |
| 0   | 0.4 | * | 2 |
| 0   | 0.8 | * | 2 |
| 1   | 0.6 | * | 2 |
| 1   | 0.2 | * | 2 |
| 0   | 0.4 | * | 2 |



# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)

25.025

11001.000001100110011001100110 x 2<sup>0</sup>

Movemos el decimal 4 lugares a la izquierda para dejar un 1 allí.

1.1001000001100110011001100110 x 2<sup>0+4</sup>

127 + 4 = 131

Hay sesgos establecidos para la precisión simple y doble. El sesgo de exponente para la precisión simple es 127, lo que significa que debemos agregarle el exponente de base 2 que se encontró anteriormente. Por lo tanto, el exponente que usará es 127 + 4, que es 131.

# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)

$$127 + \textcolor{red}{4} = 131$$

|     |    |    |    |   |   |   |   |   |   |
|-----|----|----|----|---|---|---|---|---|---|
| 131 |    |    |    |   |   |   |   | / | 2 |
| 1   | 65 |    |    |   |   |   |   | / | 2 |
|     | 1  | 32 |    |   |   |   |   | / | 2 |
|     |    | 0  | 16 |   |   |   |   | / | 2 |
|     |    |    | 0  | 8 |   |   |   | / | 2 |
|     |    |    |    | 0 | 4 |   |   | / | 2 |
|     |    |    |    |   | 0 | 2 | 2 | / | 2 |
|     |    |    |    |   |   | 0 | 1 | / | 2 |
| 1   |    | 1  | 0  | 0 | 0 | 0 | 0 | 1 |   |

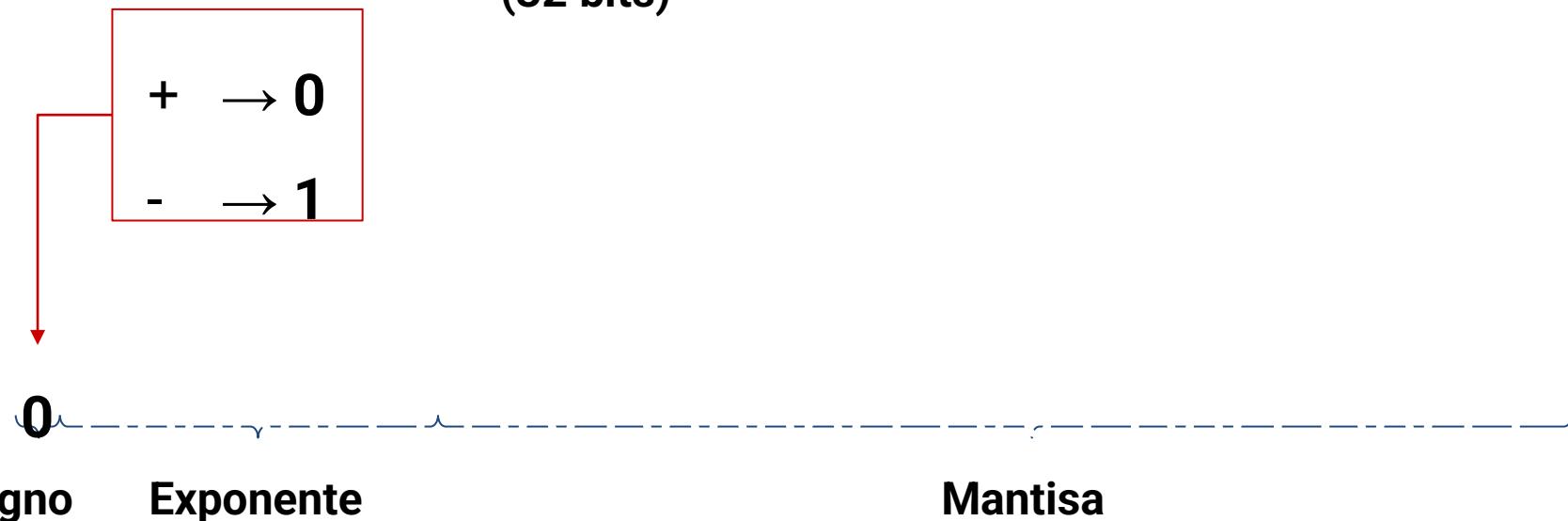
$$131 = 10000011$$

# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)

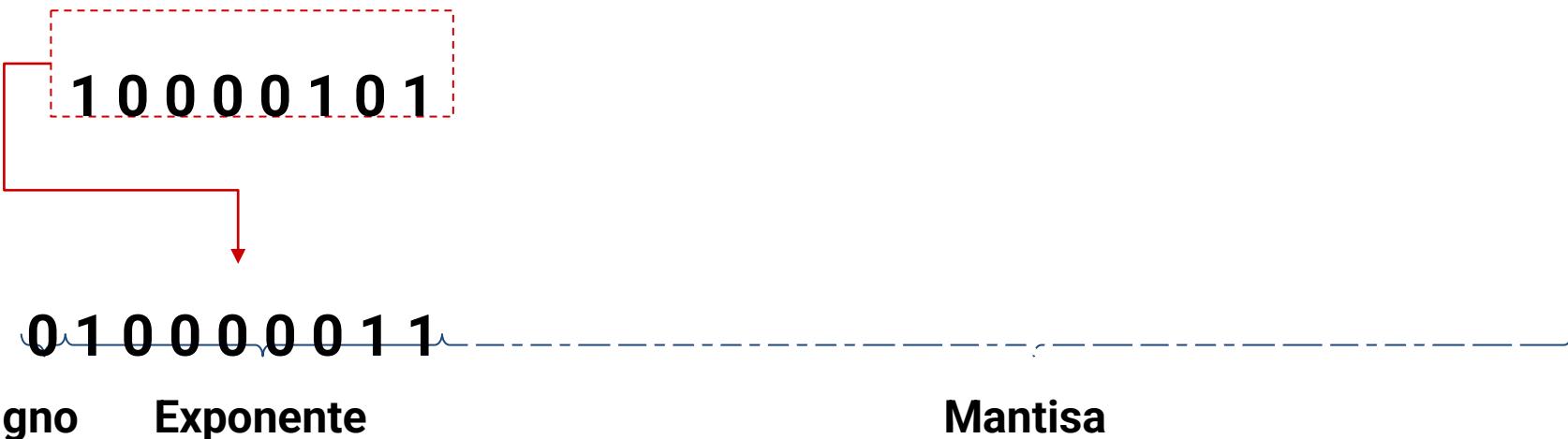


# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de coma flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)

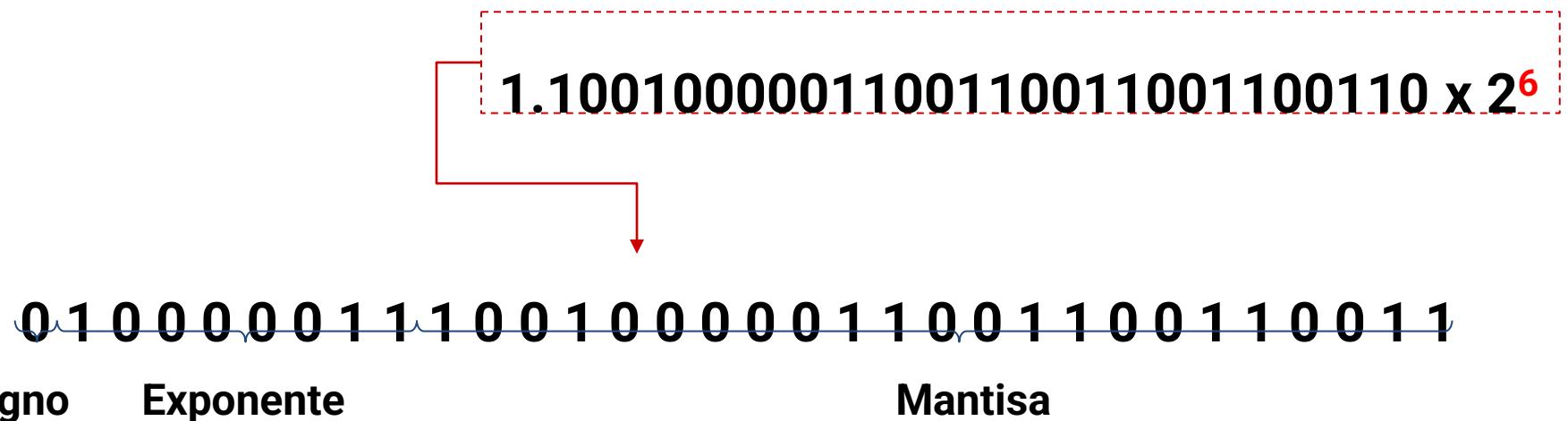


# Representación numérica (flotante)

Convertir 25.025 a binario usando una sola representación de punto flotante IEEE 754

¿Qué precisión debemos usar?

(32 bits)



Simplemente soltamos el 1 a la izquierda y copiamos la parte decimal del número que se está multiplicando por 2.

# Representación numérica (flotante)

¿Convertir el número representado en un solo coma flotante IEEE 754 a decimal?

**11000010100010000000000000000000**

# Representación numérica (flotante)

¿Convertir el número representado en un solo punto flotante IEEE 754 a decimal?

**11000010100010000000000000000000**

**Dividimos el número en sus partes (signo, exponente y mantisa).**

# Representación numérica (flotante)

¿Convertir el número representado en un solo coma flotante IEEE 754 a decimal?

11000010100010000000000000000000

**El número es negativo.**



The diagram shows the binary representation of a floating-point number: 11000010100010000000000000000000. A red square highlights the first bit (the sign bit). Below the number, three labels with blue curly braces identify the components: "Signo" covers the first bit, "Exponente" covers the next 8 bits, and "Mantisa" covers the remaining 52 bits.

11000010100010000000000000000000

Signo      Exponente      Mantisa

Identificamos el signo: 0 → positivo y 1 → negativo.

# Representación numérica (flotante)

¿Convertir el número representado en un solo coma flotante IEEE 754 a decimal?

**1100001010001000000000000000000000**

Transformamos el exponente a decimal para encontrar la normalización de la mantisa.

**Calculamos el exponente →  $10000101 = 133$**

$$133 - 127 = 6$$

# Representación numérica (flotante)

¿Convertir el número representado en un solo coma flotante IEEE 754 a decimal?

11000010100010000000000000000000

**Calculamos el número desnormalizando la mantisa y convirtiéndola a decimal.**

1.1000010100010000000000000000000  
Signo Exponete Mantisa

1.0001000000000000000000000000000

Movemos la señal decimal 6 lugares a la derecha.

**Siempre debe insertar 1 a la izquierda.**

# Representación numérica (flotante)

¿Convertir el número representado en un solo coma flotante IEEE 754 a decimal?

**1100001010001000000000000000000000**

**Calculamos el número desnormalizando la mantisa y convirtiéndola a decimal.**

**110000101000100000000000000000000000000000000000000000000**

# **Signo Exponete**

Mantisa

## Convertimos a decimal.

$$\begin{array}{r} 1000100 = 68 \\ 0000000000000000 = 0 \end{array} \quad \left. \begin{array}{l} \\ \end{array} \right\} - 68.0$$

# Representación de codificación

# Representación de codificación

## Codificaciones decimales y alfanuméricas

Una codificación es un conjunto de cadenas de n bits sobre las cuales se establece una convención en la que cada cadena representa un número u otro tipo de información..

- La codificación numérica o código representa información numérica.
- La codificación alfanumérica o código representa números, letras y signos de puntuación.
- Codificación de errores o información de código para que ciertos errores en el almacenamiento, recuperación o transmisión de información puedan detectarse y corregirse.

# Representación de codificación

## Decimal codificado binario

**Binary Coded Decimal (BCD)** se utiliza para la representación binaria de números en base decimal. Cada dígito decimal está representado por una combinación de 4 bits y cada número como una cadena de dígitos.

$$734 = 0111\ 0011\ 0100$$

$$22 = 0010\ 0010$$

| Decimal | Binay (BCD) |   |   |   |
|---------|-------------|---|---|---|
|         | 8           | 4 | 2 | 1 |
| 0       | 0           | 0 | 0 | 0 |
| 1       | 0           | 0 | 0 | 1 |
| 2       | 0           | 0 | 1 | 0 |
| 3       | 0           | 0 | 1 | 1 |
| 4       | 0           | 1 | 0 | 0 |
| 5       | 0           | 1 | 0 | 1 |
| 6       | 0           | 1 | 1 | 0 |
| 7       | 0           | 1 | 1 | 1 |
| 8       | 1           | 0 | 0 | 0 |
| 9       | 1           | 0 | 0 | 1 |

# Representación de codificación

## Códigos alfanuméricos

**Códigos alfanuméricos** se utilizan para representar texto donde se asigna un código (cadena de bits) a cada carácter. Los personajes generalmente se agrupan en 5 categorías:

- Caracteres alfanuméricos: A, B, C, ....Z, a, b, c, ...., z
- Caracteres numéricos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Caracteres especiales: ( ) + - , & > < Ñ ñ # Ç ç SP ...
- Caracteres geométricos y gráficos: | ☰¶
- Personajes de control: entrar, espacio, ...

# Representación de codificación

## Código Ascii

**ASCII (American Standard Code for Information Interchange) code** es uno de los más antiguos (1968). Fue creado para representar los caracteres y símbolos del idioma inglés. El código ASCII básico utiliza 7 bits (cada carácter o símbolo está representado por 7 bits).

**C/Vega,7**

|          |          |          |          |          |          |         |          |
|----------|----------|----------|----------|----------|----------|---------|----------|
| <b>C</b> | <b>/</b> | <b>v</b> | <b>e</b> | <b>g</b> | <b>a</b> | ,       | <b>7</b> |
| 1000011  | 0101111  | 1010110  | 1100101  | 1100111  | 1100001  | 0101100 | 0110111  |

Corresponde a la estandarización ANSI x 3.4 - 1968 o ISO 646.

|          | 0   | 1           | 2     | 3 | 4 | 5 | 6 | 7   |
|----------|-----|-------------|-------|---|---|---|---|-----|
| <b>0</b> | NUL | DLE         | space | 0 | @ | P | ` | p   |
| <b>1</b> | SOH | DC1<br>XON  | !     | 1 | A | Q | a | q   |
| <b>2</b> | STX | DC2         | "     | 2 | B | R | b | r   |
| <b>3</b> | ETX | DC3<br>XOFF | #     | 3 | C | S | c | s   |
| <b>4</b> | EOT | DC4         | \$    | 4 | D | T | d | t   |
| <b>5</b> | ENQ | NAK         | %     | 5 | E | U | e | u   |
| <b>6</b> | ACK | SYN         | &     | 6 | F | V | f | v   |
| <b>7</b> | BEL | ETB         | '     | 7 | G | W | g | w   |
| <b>8</b> | BS  | CAN         | (     | 8 | H | X | h | x   |
| <b>9</b> | HT  | EM          | )     | 9 | I | Y | i | y   |
| <b>A</b> | LF  | SUB         | *     | : | J | Z | j | z   |
| <b>B</b> | VT  | ESC         | +     | ; | K | [ | k | {   |
| <b>C</b> | FF  | FS          | ,     | < | L | \ | l |     |
| <b>D</b> | CR  | GS          | -     | = | M | ] | m | }   |
| <b>E</b> | SO  | RS          | .     | > | N | ^ | n | ~   |
| <b>F</b> | SI  | US          | /     | ? | O | _ | o | del |

# Representación de codificación

## Código Ascii

**ASCII (American Standard Code for Information Interchange) code** es uno de los más antiguos (1968). Fue creado para representar los caracteres y símbolos del idioma inglés. El código ASCII básico utiliza 7 bits (cada carácter o símbolo está representado por 7 bits).

**C/Vega,7**

|          |          |          |          |          |          |         |          |
|----------|----------|----------|----------|----------|----------|---------|----------|
| <b>C</b> | <b>/</b> | <b>v</b> | <b>e</b> | <b>g</b> | <b>a</b> | ,       | <b>7</b> |
| 1000011  | 0101111  | 1010110  | 1100101  | 1100111  | 1100001  | 0101100 | 0110111  |
| 43       | 2F       | 56       | 65       | 67       | 61       | 2C      | 37       |

Corresponde a la estandarización ANSI x 3.4 - 1968 o ISO 646.

| Decimal | Binary | Hexadecimal |
|---------|--------|-------------|
| 0       | 0000   | 0           |
| 1       | 0001   | 1           |
| 2       | 0010   | 2           |
| 3       | 0011   | 3           |
| 4       | 0100   | 4           |
| 5       | 0101   | 5           |
| 6       | 0110   | 6           |
| 7       | 0111   | 7           |
| 8       | 1000   | 8           |
| 9       | 1001   | 9           |
| 10      | 1010   | A           |
| 11      | 1011   | B           |
| 12      | 1100   | C           |
| 13      | 1101   | D           |
| 14      | 1110   | E           |
| 15      | 1111   | F           |

# Representación de codificación

## Código Ascii

Existen diferentes versiones extendidas del código ascii usando 8 bits.

| Name           | ISO family | Geographical area                      |
|----------------|------------|----------------------------------------|
| Latin-1        | ISO 8859-1 | Western and Eastern Europe             |
| Latin-2        | ISO 8859-2 | Central and Eastern Europe             |
| Latin-3        | ISO 8859-3 | Southern Europe, Maltese and Esperanto |
| Latin-4        | ISO 8859-4 | North europe                           |
| Latin/cyrillic | ISO 8859-5 | Slavic languages                       |
| Latin/arabic   | ISO 8859-6 | Arabic languages                       |
| Latin/greek    | ISO 8859-7 | Modern greek                           |

ISO/IEC 8859 es una serie conjunta de normas ISO e IEC para codificaciones de caracteres de 8 bits.

# Representación de codificación

## Código Ascii extendido

ASCII extendido utiliza una codificación de caracteres de ocho bits que incluye (la mayoría de) los caracteres ASCII de siete bits, además de caracteres adicionales.

| ASCII control characters |                           |    |       | ASCII printable characters |   |     |   | Extended ASCII characters |   |     |      |
|--------------------------|---------------------------|----|-------|----------------------------|---|-----|---|---------------------------|---|-----|------|
| 00                       | NULL (Null character)     | 32 | space | 64                         | @ | 96  | ` | 128                       | ç | 160 | á    |
| 01                       | SOH (Start of Header)     | 33 | !     | 65                         | A | 97  | a | 129                       | ü | 161 | í    |
| 02                       | STX (Start of Text)       | 34 | "     | 66                         | B | 98  | b | 130                       | é | 162 | ó    |
| 03                       | ETX (End of Text)         | 35 | #     | 67                         | C | 99  | c | 131                       | â | 163 | ú    |
| 04                       | EOT (End of Trans.)       | 36 | \$    | 68                         | D | 100 | d | 132                       | ä | 164 | ñ    |
| 05                       | ENQ (Enquiry)             | 37 | %     | 69                         | E | 101 | e | 133                       | à | 165 | À    |
| 06                       | ACK (Acknowledgement)     | 38 | &     | 70                         | F | 102 | f | 134                       | à | 166 | à    |
| 07                       | BEL (Bell)                | 39 | '     | 71                         | G | 103 | g | 135                       | ç | 167 | º    |
| 08                       | BS (Backspace)            | 40 | (     | 72                         | H | 104 | h | 136                       | é | 168 | ¿    |
| 09                       | HT (Horizontal Tab)       | 41 | )     | 73                         | I | 105 | i | 137                       | ë | 169 | ®    |
| 10                       | LF (Line feed)            | 42 | *     | 74                         | J | 106 | j | 138                       | ê | 170 | »    |
| 11                       | VT (Vertical Tab)         | 43 | +     | 75                         | K | 107 | k | 139                       | í | 171 | ½    |
| 12                       | FF (Form feed)            | 44 | ,     | 76                         | L | 108 | l | 140                       | í | 172 | ¼    |
| 13                       | CR (Carriage return)      | 45 | -     | 77                         | M | 109 | m | 141                       | í | 173 | í    |
| 14                       | SO (Shift Out)            | 46 | .     | 78                         | N | 110 | n | 142                       | Ã | 174 | «    |
| 15                       | SI (Shift In)             | 47 | /     | 79                         | O | 111 | o | 143                       | Á | 175 | »    |
| 16                       | DLE (Data link escape)    | 48 | 0     | 80                         | P | 112 | p | 144                       | É | 176 | „    |
| 17                       | DC1 (Device control 1)    | 49 | 1     | 81                         | Q | 113 | q | 145                       | æ | 177 | „    |
| 18                       | DC2 (Device control 2)    | 50 | 2     | 82                         | R | 114 | r | 146                       | Æ | 178 | “    |
| 19                       | DC3 (Device control 3)    | 51 | 3     | 83                         | S | 115 | s | 147                       | ó | 179 | —    |
| 20                       | DC4 (Device control 4)    | 52 | 4     | 84                         | T | 116 | t | 148                       | ö | 180 | —    |
| 21                       | NAK (Negative acknowl.)   | 53 | 5     | 85                         | U | 117 | u | 149                       | ò | 181 | À    |
| 22                       | SYN (Synchronous idle)    | 54 | 6     | 86                         | V | 118 | v | 150                       | ú | 182 | Á    |
| 23                       | ETB (End of trans. block) | 55 | 7     | 87                         | W | 119 | w | 151                       | û | 183 | À    |
| 24                       | CAN (Cancel)              | 56 | 8     | 88                         | X | 120 | x | 152                       | ÿ | 184 | ®    |
| 25                       | EM (End of medium)        | 57 | 9     | 89                         | Y | 121 | y | 153                       | Ó | 185 | —    |
| 26                       | SUB (Substitute)          | 58 | :     | 90                         | Z | 122 | z | 154                       | Ù | 186 | —    |
| 27                       | ESC (Escape)              | 59 | :     | 91                         | [ | 123 | { | 155                       | ø | 187 | —    |
| 28                       | FS (File separator)       | 60 | <     | 92                         | \ | 124 |   | 156                       | £ | 188 | —    |
| 29                       | GS (Group separator)      | 61 | =     | 93                         | ] | 125 | } | 157                       | Ø | 189 | ¢    |
| 30                       | RS (Record separator)     | 62 | >     | 94                         | ^ | 126 | ~ | 158                       | x | 190 | ¥    |
| 31                       | US (Unit separator)       | 63 | ?     | 95                         | - |     |   | 191                       | ƒ | 192 | nbsp |
| 127                      | DEL (Delete)              |    |       |                            |   |     |   | 193                       | — | 194 | —    |

| Dec | Hex | Char |
|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 128 | 80  | ç    | 160 | A0  | á    | 192 | C0  | ł    | 224 | E0  | ą    |
| 129 | 81  | ü    | 161 | A1  | í    | 193 | C1  | ł    | 225 | E1  | ő    |
| 130 | 82  | é    | 162 | ó   | —    | 194 | C2  | ł    | 226 | E2  | ѓ    |
| 131 | 83  | å    | 163 | ú   | —    | 195 | C3  | ł    | 227 | E3  | π    |
| 132 | 84  | ä    | 164 | A4  | ñ    | 196 | C4  | —    | 228 | E4  | ζ    |
| 133 | 85  | à    | 165 | A5  | ń    | 197 | C5  | —    | 229 | E5  | σ    |
| 134 | 86  | ä    | 166 | A6  | —    | 198 | C6  | —    | 230 | E6  | μ    |
| 135 | 87  | œ    | 167 | A7  | —    | 199 | C7  | —    | 231 | E7  | ı    |
| 136 | 88  | ö    | 168 | A8  | —    | 200 | C8  | —    | 232 | E8  | ø    |
| 137 | 89  | ë    | 169 | A9  | —    | 201 | C9  | —    | 233 | E9  | ø    |
| 138 | 8A  | è    | 170 | AA  | —    | 202 | CA  | —    | 234 | EA  | Ω    |
| 139 | 8B  | í    | 171 | AB  | ½    | 203 | CB  | —    | 235 | EB  | đ    |
| 140 | 8C  | —    | 172 | AC  | ¼    | 204 | CC  | —    | 236 | EC  | »    |
| 141 | 8D  | —    | 173 | AD  | —    | 205 | CD  | —    | 237 | ED  | ♀    |
| 142 | 8E  | —    | 174 | AE  | —    | 206 | CE  | —    | 238 | EE  | €    |
| 143 | 8F  | —    | 175 | AF  | —    | 207 | CF  | —    | 239 | EF  | □    |
| 144 | 90  | —    | 176 | B0  | —    | 208 | DD  | —    | 240 | FO  | ≡    |
| 145 | 91  | —    | 177 | B1  | —    | 209 | D1  | —    | 241 | F1  | ±    |
| 146 | 92  | —    | 178 | B2  | —    | 210 | D2  | —    | 242 | F2  | ≥    |
| 147 | 93  | —    | 179 | B3  | —    | 211 | D3  | —    | 243 | F3  | ≤    |
| 148 | 94  | —    | 180 | B4  | —    | 212 | D4  | —    | 244 | F4  | —    |
| 149 | 95  | —    | 181 | B5  | —    | 213 | D5  | —    | 245 | F5  | —    |
| 150 | 96  | —    | 182 | B6  | —    | 214 | D6  | —    | 246 | F6  | —    |
| 151 | 97  | —    | 183 | B7  | —    | 215 | D7  | —    | 247 | F7  | —    |
| 152 | 98  | —    | 184 | B8  | —    | 216 | D8  | —    | 248 | F8  | —    |
| 153 | 99  | —    | 185 | B9  | —    | 217 | D9  | —    | 249 | F9  | —    |
| 154 | 9A  | —    | 186 | BA  | —    | 218 | DA  | —    | 250 | FA  | —    |
| 155 | 9B  | —    | 187 | BB  | —    | 219 | DB  | —    | 251 | FB  | —    |
| 156 | 9C  | —    | 188 | BC  | —    | 220 | DC  | —    | 252 | FC  | —    |
| 157 | 9D  | —    | 189 | BD  | —    | 221 | DD  | —    | 253 | FD  | —    |
| 158 | 9E  | —    | 190 | BE  | —    | 222 | DE  | —    | 254 | FE  | —    |
| 159 | 9F  | —    | 191 | BF  | —    | 223 | DF  | —    | 255 | FF  | —    |



# Representación de codificación

## Del texto al binario

| H        | e        | I        | I        | o        |          | w        | o        | r        | I        | d        |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 01001000 | 01100101 | 01101100 | 01101100 | 01101111 | 00100000 | 01010111 | 01101111 | 01110010 | 01101100 | 01100100 |
| 48       | 65       | 6C       | 6C       | 6F       | 20       | 57       | 6F       | 72       | 6C       | 64       |

La conversión de caracteres a binarios se simplifica cuando se convierten primero a hexadecimales y, posteriormente, se transforman en binarios.

| Dec | Hex | Char |
|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|
| 128 | 80  | ç    | 160 | A0  | á    | 192 | C0  | ł    | 224 | E0  | ¤    |
| 129 | 81  | ü    | 161 | A1  | í    | 193 | C1  | ł    | 225 | E1  | ß    |
| 130 | 82  | é    | 162 | A2  | ó    | 194 | C2  | ł    | 226 | E2  | ѓ    |
| 131 | 83  | ã    | 163 | A3  | ú    | 195 | C3  | ł    | 227 | E3  | ŉ    |
| 132 | 84  | ä    | 164 | A4  | ñ    | 196 | C4  | —    | 228 | E4  | Σ    |
| 133 | 85  | ä    | 165 | A5  | Ñ    | 197 | C5  | +    | 229 | E5  | σ    |
| 134 | 86  | ä    | 166 | A6  | æ    | 198 | C6  | —    | 230 | E6  | µ    |
| 135 | 87  | ç    | 167 | A7  | ø    | 199 | C7  | —    | 231 | E7  | ı    |
| 136 | 88  | ø    | 168 | A8  | Ը    | 200 | C8  | —    | 232 | E8  | Փ    |
| 137 | 89  | é    | 169 | A9  | ܵ    | 201 | C9  | ܵ    | 233 | E9  | ܵ    |
| 138 | 8A  | è    | 170 | AA  | ܵ    | 202 | CA  | ܵ    | 234 | EA  | ܵ    |
| 139 | 8B  | ܵ    | 171 | AB  | ܵ    | 203 | CB  | ܵ    | 235 | EB  | ܵ    |
| 140 | 8C  | ܵ    | 172 | AC  | ܵ    | 204 | CC  | ܵ    | 236 | EC  | ܵ    |
| 141 | 8D  | ܵ    | 173 | AD  | ܵ    | 205 | CD  | ܵ    | 237 | ED  | ܵ    |
| 142 | 8E  | ܵ    | 174 | AE  | ܵ    | 206 | CE  | ܵ    | 238 | EE  | ܵ    |
| 143 | 8F  | ܵ    | 175 | AF  | ܵ    | 207 | CF  | ܵ    | 239 | EF  | ܵ    |
| 144 | 90  | ܵ    | 176 | B0  | ܵ    | 208 | DO  | ܵ    | 240 | F0  | ܵ    |
| 145 | 91  | ܵ    | 177 | B1  | ܵ    | 209 | D1  | ܵ    | 241 | F1  | ܵ    |
| 146 | 92  | ܵ    | 178 | B2  | ܵ    | 210 | D2  | ܵ    | 242 | F2  | ܵ    |
| 147 | 93  | ܵ    | 179 | B3  | ܵ    | 211 | D3  | ܵ    | 243 | F3  | ܵ    |
| 148 | 94  | ܵ    | 180 | B4  | ܵ    | 212 | D4  | ܵ    | 244 | F4  | ܵ    |
| 149 | 95  | ܵ    | 181 | B5  | ܵ    | 213 | D5  | ܵ    | 245 | F5  | ܵ    |
| 150 | 96  | ܵ    | 182 | B6  | ܵ    | 214 | D6  | ܵ    | 246 | F6  | ܵ    |
| 151 | 97  | ܵ    | 183 | B7  | ܵ    | 215 | D7  | ܵ    | 247 | F7  | ܵ    |
| 152 | 98  | ܵ    | 184 | B8  | ܵ    | 216 | D8  | ܵ    | 248 | F8  | ܵ    |
| 153 | 99  | ܵ    | 185 | B9  | ܵ    | 217 | D9  | ܵ    | 249 | F9  | ܵ    |
| 154 | 9A  | ܵ    | 186 | BA  | ܵ    | 218 | DA  | ܵ    | 250 | FA  | ܵ    |
| 155 | 9B  | ܵ    | 187 | BB  | ܵ    | 219 | DB  | ܵ    | 251 | FB  | ܵ    |
| 156 | 9C  | ܵ    | 188 | BC  | ܵ    | 220 | DC  | ܵ    | 252 | FC  | ܵ    |
| 157 | 9D  | ܵ    | 189 | BD  | ܵ    | 221 | DD  | ܵ    | 253 | FD  | ܵ    |
| 158 | 9E  | ܵ    | 190 | BE  | ܵ    | 222 | DE  | ܵ    | 254 | FE  | ܵ    |
| 159 | 9F  | ܵ    | 191 | BF  | ܵ    | 223 | DF  | ܵ    | 255 | FF  | ܵ    |

# Representación de codificación

## Unicode

El estándar **Unicode** es un estándar de información para la codificación, representación y manejo consistentes del texto expresado en la mayoría de los sistemas de escritura del mundo. Este código fue diseñado para seguir estas propiedades principales:

- Universalidad: cubre la mayoría de las lenguas escritas existentes.
- Singularidad: Cada símbolo tiene un código único.
- Uniformidad: Cada carácter está representado por 8 o 16 bits dependiendo de la versión unicode.

# Representación de codificación

## Unicode

Los códigos se dividen en 4 grupos o zonas, como se muestra en la tabla.

| Zone | Codes (HEX) | Symbols                                                                                                                                                      | Characters |
|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| A    | 0000 - 3FFF | Basic Latin (ASCII), Latin-1 and other Latin characters, Greek, Cyrillic, Armenian, Hebrew, Arabic, Syrian, Chinese, Japanese and Korean phonetic characters | 8192       |
| I    | 4000 - 9FFF | Chinese, Japanese and Korean ideograms                                                                                                                       | 24576      |
| O    | A000 - DFFF | Not assigned                                                                                                                                                 | 16384      |
| R    | E000 - FFFF | Local and user-specific characters                                                                                                                           | 8192       |

# Anexos

Base change

<https://youtu.be/5WtLFbriEEE>

Integer numbers

<https://youtu.be/B7SpmkW0lTs>

Two's complement

<https://youtu.be/UTVuROxztuQ>

IEEE (Floating point)

<https://youtu.be/HcjXH9WGmAU>

Some tips

<https://youtu.be/5TIUWLxOWzU>

# **Fundamentos de Ingeniería Informática**

## **Modulo III - Unidad 10 Bases de datos**

Profesor: Héctor Molina García



Universidad  
Francisco de Vitoria  
**UFV** Madrid

*Grado en Ingeniería Informática  
Escuela Politécnica Superior*

# Conceptos básicos sobre las bases de datos.

# Conceptos básicos sobre las bases de datos

Datos



vs

Información



Los datos son elementos crudos, desorganizados y sin refinar.

La información es la organización e interpretación de esos elementos.

# Conceptos básicos sobre las bases de datos

La información se clasifica en función de cómo se almacenan y organizan los datos a través de algún tipo de estructura y/o etiquetado.

# Conceptos básicos sobre las bases de datos

La información se clasifica en función de cómo se almacenan y organizan los datos a través de algún tipo de estructura y/o etiquetado.

## Estructurado

La información se organiza utilizando algún modelo de datos o esquema. Hay una definición precisa del significado de cada elemento.  
Hay una estructura definida e identifiable.

## Databases

# Conceptos básicos sobre las bases de datos

La información se clasifica en función de cómo se almacenan y organizan los datos a través de algún tipo de estructura y/o etiquetado.

## Estructurado

La información se organiza utilizando algún modelo de datos o esquema. Hay una definición precisa del significado de cada elemento.  
Hay una estructura definida e identifiable.

## Semiestructurado

La información no obedece a la estructura tabular de los modelos de datos asociados con bases de datos relacionales u otras formas de tablas de datos.  
Hay etiquetas u otros marcadores que incluyen cierta semántica.

**Databases**

**XML, HTML, JSON**

# Conceptos básicos sobre las bases de datos

La información se clasifica en función de cómo se almacenan y organizan los datos a través de algún tipo de estructura y/o etiquetado.

## Estructurado

La información se organiza utilizando algún modelo de datos o esquema. Hay una definición precisa del significado de cada elemento. Hay una estructura definida e identifiable.

**Databases**

## Semiestructurado

La información no obedece a la estructura tabular de los modelos de datos asociados con bases de datos relacionales u otras formas de tablas de datos. Hay etiquetas u otros marcadores que incluyen cierta semántica.

**XML, HTML, JSON**

## Desestructurado

Los datos no estructurados son información que no está organizada de acuerdo con un modelo o esquema de datos preestablecido.

**Texto sin formato**

# Conceptos básicos sobre las bases de datos

Existen diferentes sistemas de almacenamiento en función del formato que queramos utilizar para almacenar los datos.

## Archivos

Archivos de texto plano con información útil.

Por lo general, almacenan datos sin procesar.

La información es desordenada y desestructurada.

# Conceptos básicos sobre las bases de datos

Existen diferentes sistemas de almacenamiento en función del formato que queramos utilizar para almacenar los datos.

## Archivos

Archivos de texto plano con información útil.  
Por lo general, almacenan datos sin procesar.  
La información es desordenada y desestructurada.

## Archivos distribuidos

Almacenamiento masivo en archivos de datos estructurados.  
Se almacenan en un entorno distribuido.  
La información tiene cierta estructura.

# Conceptos básicos sobre las bases de datos

Existen diferentes sistemas de almacenamiento en función del formato que queramos utilizar para almacenar los datos.

## Archivos

Archivos de texto plano con información útil.  
Por lo general, almacenan datos sin procesar.  
La información es desordenada y desestructurada.

## Archivos distribuidos

Almacenamiento masivo en archivos de datos estructurados.  
Se almacenan en un entorno distribuido.  
La información tiene cierta estructura.

## Base de datos

Almacenamiento en bases de datos SQL y NOSQL.

# Bases de datos

# Bases de datos

Una base de datos es una colección de información almacenada en una computadora o sistema informático en una forma a la que se puede acceder, recuperar y modificar fácilmente..

- Facilitar el almacenamiento de grandes cantidades de información.
- Facilitar la recuperación de información de forma rápida y flexible.
- Facilitar la organización permitiendo la vinculación de diferentes tipos de información
- Facilitar la impresión y distribución de información en una amplia variedad de formas.

# Bases de datos

Existen diferentes tipos de bases de datos, dependiendo de:

Cómo se organiza la información

- Bases de datos SQL

- Bases de datos NoSQL (no solo SQL)

Cómo se almacena la información en el nivel físico

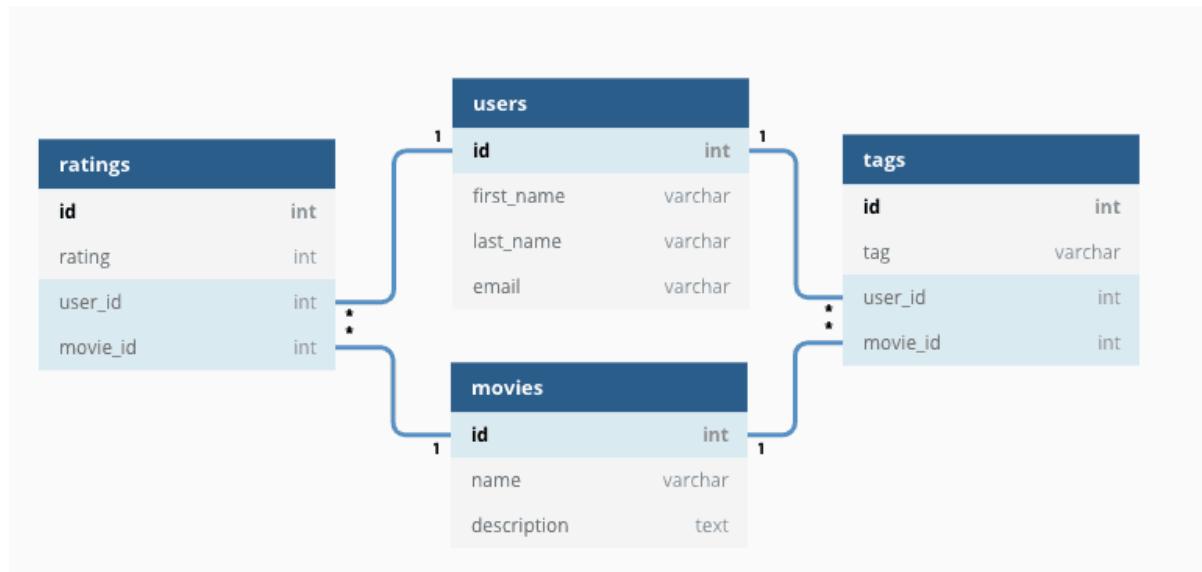
- Centralizada

- Distribuido

- Nube

# Bases de datos

Una base de datos relacional (RDB) es una forma de estructurar información en tablas, filas y columnas. Un RDB tiene la capacidad de establecer vínculos (o relaciones) entre la información mediante la unión de tablas, lo que facilita la comprensión y la obtención de información sobre la relación entre varios puntos de datos.



A menudo, una base de datos relacional se puede denominar base de datos SQL.

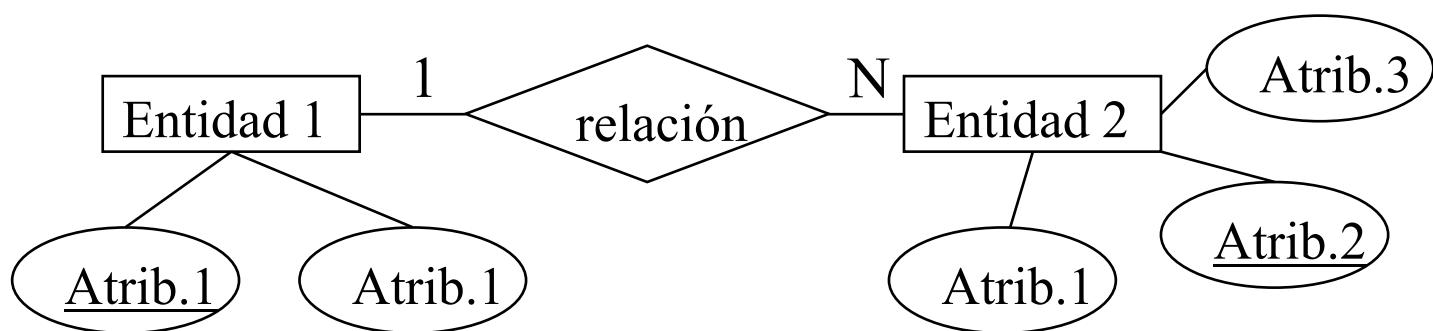
# Bases de datos

Las bases de datos relacionales se describen utilizando un diagrama entidad-relación que se compone de tres elementos básicos:

- **Entidad:** cosa, persona, lugar, unidad, objeto o cualquier elemento cuya información se almacene. Por ejemplo, los usuarios del diagrama anterior.
- **Atributo:** Es información específica (características) de una entidad. Son las propiedades de la entidad. Por ejemplo, el nombre de pila de un usuario.
- **Clave primaria:** Es un atributo especial que identifica cada registro en una entidad.
- **Relación:** son vínculos o relaciones, como su nombre indica, entre entidades. Por ejemplo, un usuario específico puede calificar varias películas de acuerdo con el diagrama anterior..

# Bases de datos

- Representación de un diagrama E-R
  - Las Entidades se representan con rectángulos
  - Sus Atributos con óvalos, aquellos que constituyen una clave primaria se subrayan.
  - Las Relaciones entre entidades con rombos. Los extremos de la relación se etiquetan para indicar el tipo de esta:
    - Un 1 en un extremo indica que la relación se establece entre una única entidad de ese extremo
    - Una N (o M) indica que la relación se establece con más de una entidad de ese extremo



# Bases de datos

## TIPOS DE RELACIONES

- Dados dos conjuntos de entidades A y B, las relaciones entre ambos pueden ser:
  - Una relación **uno a uno (1:1)**: Una entidad A se relaciona únicamente con una entidad B y viceversa. Esta relación no es muy común, porque a menudo una de las entidades se define como atributo de la otra.
    - Ej: la relación coche-matrícula. Cada coche tiene una única matrícula y cada matrícula pertenece únicamente a un coche
  - Relación **uno a varios (1:N)**: Una entidad en A se relaciona con cero o muchas entidades en B. Pero una entidad en B se relaciona con una única entidad en A.
    - Ej: la relación entre Clientes y Pedidos. Un cliente puede realizar cualquier número de pedidos. Pero cada pedido es realizado únicamente por un cliente.
  - Una relación de **varios a varios (N:M)**: Una entidad en A se puede relacionar con 0 o muchas entidades en B y viceversa.
    - Ejemplo: la relación entre Productos y Pedidos. Un solo pedido puede incluir varios productos. Por otro lado, un único producto puede aparecer en muchos pedidos.

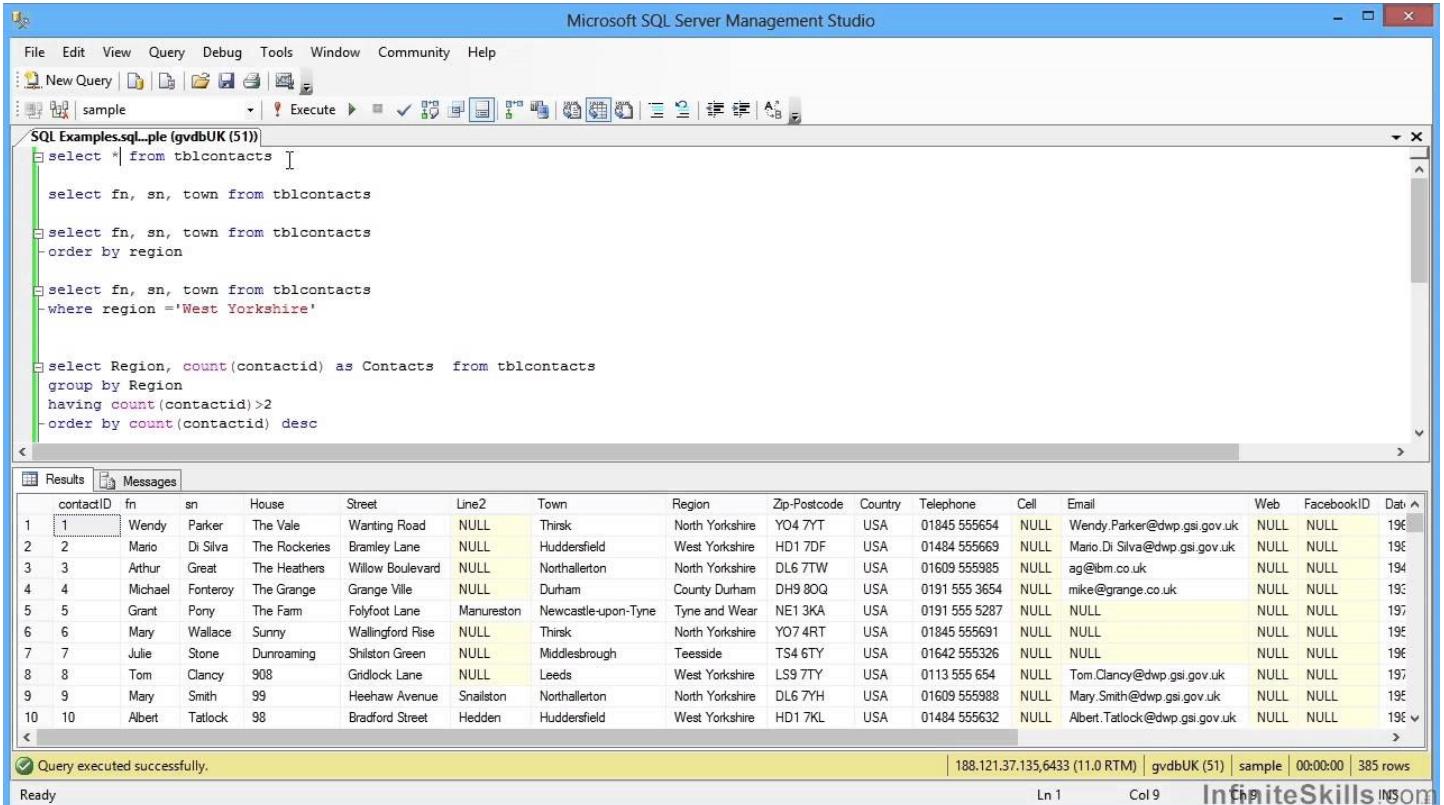
# Bases de datos

**DataBase Management System (DBMS)** es un software de aplicación diseñado para almacenar, recuperar, consultar y administrar datos. Los DBMS proporcionan algunas funciones que permiten la administración de una base de datos y sus datos:

- Definición: Creación, modificación y eliminación de definiciones que definen la organización de los datos.
- Gestión y explotación: Inserción, modificación, consulta y supresión de los datos.
- Administración: Registro y supervisión de usuarios, aplicación de la seguridad de los datos, supervisión del rendimiento, mantenimiento de la integridad de los datos, gestión del control de simultaneidad y recuperación de información que ha sido dañada por algún evento, como un error inesperado del sistema..

# Bases de datos

**SQL (Structured Query Language)** es un lenguaje de programación específico de dominio diseñado para administrar y recuperar información de sistemas de administración de bases de datos relacionales..



The screenshot shows the Microsoft SQL Server Management Studio interface. In the top-left corner, there's a toolbar with various icons for file operations like New Query, Save, Print, and Help. Below the toolbar is a menu bar with File, Edit, View, Query, Debug, Tools, Window, Community, and Help. The main area has a title bar "Microsoft SQL Server Management Studio". A tab bar at the top indicates the current database is "sample". The left pane contains a query editor window titled "SQL Examples.sql...ple (gvdbUK (51))" with several SQL statements. The first statement is "select \* from tblcontacts". The results pane below displays a table with 10 rows of contact information. The columns are: contactID, fn, sn, House, Street, Line2, Town, Region, Zip-Postcode, Country, Telephone, Cell, Email, Web, FacebookID, and Date. The data includes names like Wendy Parker, Mario Di Silva, Arthur Great, Michael Fonteroy, Grant Pony, Mary Wallace, Julie Stone, Tom Clancy, Mary Smith, and Albert Tatlock, along with their respective addresses and contact details. At the bottom of the results pane, a message says "Query executed successfully." and provides some statistics: 188.121.37.135, 6433 (11.0 RTM), gvdbUK (51), sample, 00:00:00, 385 rows.

<https://dev.mysql.com/doc/mysql-tutorial-excerpt/8.0/en/examples.html>

# Bases de datos

**SQL (Structured Query Language)** es un lenguaje de programación específico de dominio diseñado para administrar y recuperar información de sistemas de administración de bases de datos relacionales.

```
CREATE TABLE shop (
    article INT(4) UNSIGNED ZEROFILL DEFAULT '0000' NOT
NULL,
    dealer  CHAR(20)                                DEFAULT ''      NOT
NULL,
    price   DOUBLE(16,2)                            DEFAULT '0.00'  NOT
NULL,
    PRIMARY KEY(article, dealer)) ;

INSERT INTO shop VALUES
(1,'A',3.45),(1,'B',3.99),(2,'A',10.99),(3,'B',1.45),
(3,'C',1.69),(3,'D',1.25),(4,'D',19.95);

INSERT INTO shop(article, dealer, price) VALUES
(1,'A',3.45);
```

```
SELECT * FROM shop;

+-----+-----+-----+
| article | dealer | price |
+-----+-----+-----+
| 0001   | A     | 3.45  |
| 0001   | B     | 3.99  |
| 0002   | A     | 10.99 |
| 0003   | B     | 1.45  |
| 0003   | C     | 1.69  |
| 0003   | D     | 1.25  |
| 0004   | D     | 19.95 |
+-----+-----+-----+
```

# Bases de datos

**SQL (Structured Query Language)** es un lenguaje de programación específico de dominio diseñado para administrar y recuperar información de sistemas de administración de bases de datos relacionales.

```
CREATE TABLE shop (
    article INT(4) UNSIGNED ZEROFILL DEFAULT '0000' NOT
    NULL,
    dealer CHAR(20)                                DEFAULT ''      NOT
    NULL,
    price   DOUBLE(16,2)                            DEFAULT '0.00'  NOT
    NULL,
    PRIMARY KEY(article, dealer));
```

```
INSERT INTO shop VALUES
(1, 'A', 3.45), (1, 'B', 3.99), (2, 'A', 10.99), (3, 'B', 1.45),
(3, 'C', 1.69), (3, 'D', 1.25), (4, 'D', 19.95);
```

```
INSERT INTO shop(article, dealer, price) VALUES
(1, 'A', 3.45);
```

```
SELECT * FROM shop;
```

| article | dealer | price |
|---------|--------|-------|
| 0001    | A      | 3.45  |
| 0001    | B      | 3.99  |
| 0002    | A      | 10.99 |
| 0003    | B      | 1.45  |
| 0003    | C      | 1.69  |
| 0003    | D      | 1.25  |
| 0004    | D      | 19.95 |

Hay diferentes formas de insertar datos en una base de datos, pero estas son las más comunes. Aunque, la sintaxis puede cambiar según el DBMS.

**Queremos crear una base de datos para una tienda online.**

**¿Qué entidades necesitamos para almacenar los datos básicos?**

**Queremos crear una base de datos para una tienda online.**

**¿Qué entidades necesitamos para almacenar los datos básicos?**

Clientes, Productos y pedidos.

**Queremos crear una base de datos para una tienda online.**

**¿Qué entidades necesitamos para almacenar los datos básicos?**

Clientes, Productos y pedidos.

**¿Qué información necesitamos para almacenar los datos del cliente?**

**Queremos crear una base de datos para una tienda online.**

**¿Qué entidades necesitamos para almacenar los datos básicos?**

**Cientes, Productos y pedidos.**

**¿Qué información necesitamos para almacenar los datos del cliente?**

- Cliente: Nombre, Apellidos, DNI, dirección de facturación, dirección de envío y más...
- Producto: Referencia, nombre, descripción, precio, unidades disponibles y más ...
- Orden: Order\_id, fecha, client\_id, product\_id, estado y más ...

**Queremos crear una base de datos para una tienda online.**

**¿Qué entidades necesitamos para almacenar los datos básicos?**

**Cientes, Productos y pedidos.**

**¿Qué información necesitamos para almacenar los datos del cliente?**

- Cliente: Nombre, Apellidos, DNI, dirección de facturación, dirección de envío y más...
- Producto: Referencia, nombre, descripción, precio, unidades disponibles y más ...
- Orden: Order\_id, fecha, client\_id, product\_id, estado y más ...

**¿Qué relaciones necesitamos?**

# Bases de datos

**Queremos crear una base de datos para una tienda online.**

**¿Qué entidades necesitamos para almacenar los datos básicos?**

**Cientes, Productos y pedidos.**

**¿Qué información necesitamos para almacenar los datos del cliente?**

- Cliente: Nombre, Apellidos, DNI, dirección de facturación, dirección de envío y más...
- Producto: Referencia, nombre, descripción, precio, unidades disponibles y más ...
- Orden: Order\_id, fecha, client\_id, product\_id, estado y más ...

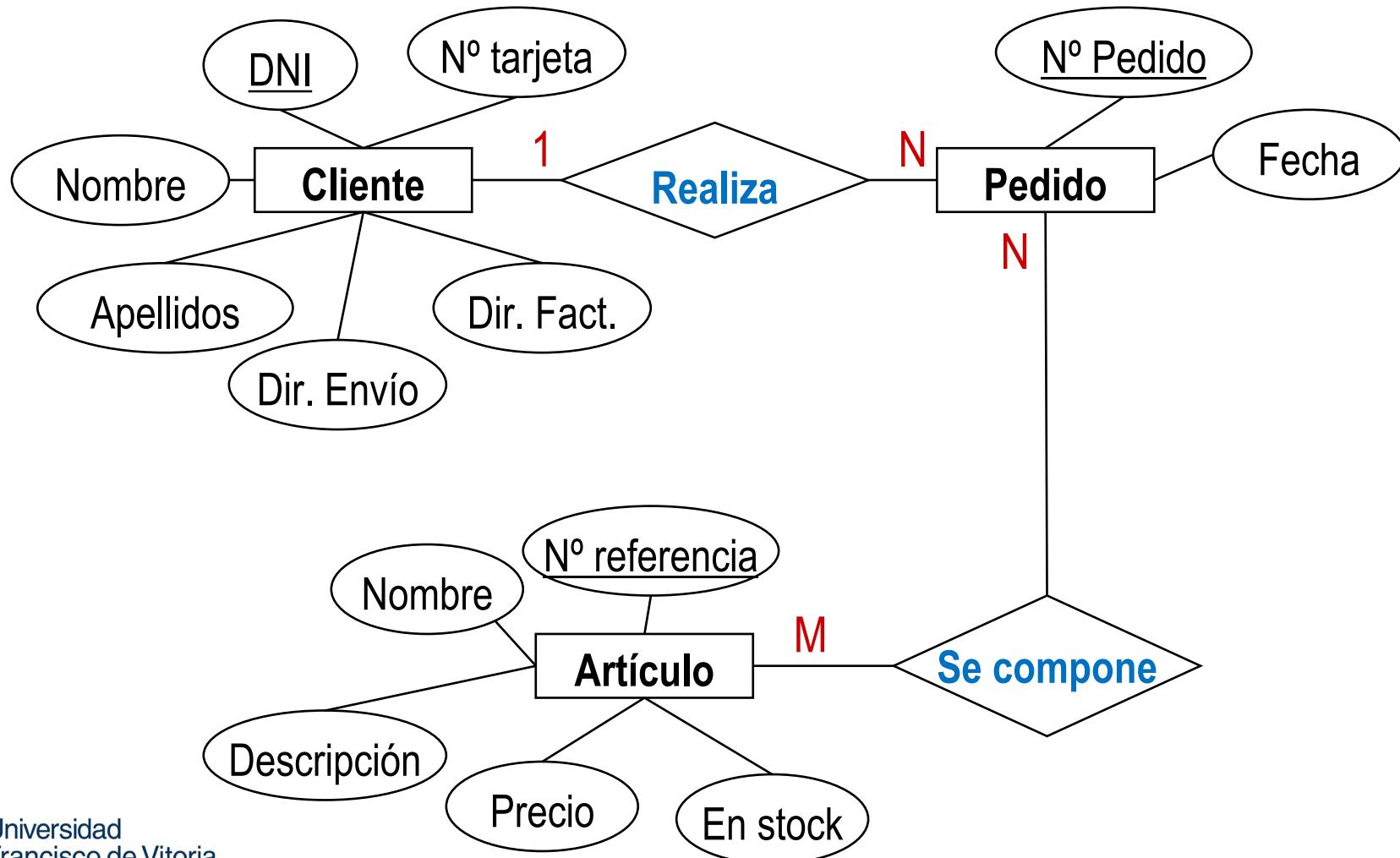
**¿Qué relaciones necesitamos?**

1:N → cliente:pedido

N:M → producto:pedido

# Bases de datos

Queremos crear una base de datos para una tienda online.



**Queremos crear una base de datos para una tienda online.**

## **Definición de la información**

- Las diferentes entidades se convierten en tablas, donde cada atributo es un campo de la tabla.

**Los campos tienen un tipo: varchar, int, double, char, etc.**

**Cada elemento concreto dentro de la tabla (fila) se llama registro.**

**Las claves primarias se crean utilizando campos que deben ser únicos para identificar cada elemento concreto.**

## Queremos crear una base de datos para una tienda online.

### Definición de las cardinalidades de las relaciones

La relación 1 a muchos (1:N) se incluye para describir cuando un cliente tiene muchos pedidos, pero cada pedido específico solo está relacionado con un cliente.

**Esta relación incluirá la clave principal del lado de la entidad 1 en el lado de la entidad N.**

**El campo DNI (clave primaria del cliente) se incluye como campo de la tabla de pedidos. Este nuevo campo se denomina clave externa.**

UNA CLAVE EXTERNA es un campo (o colección de campos) en una tabla, que hace referencia a la CLAVE PRIMARIA en otra tabla.

## Queremos crear una base de datos para una tienda online.

### Definición de las cardinalidades de las relaciones

La relación muchos a muchos (N:M) se incluye para describir cuándo un pedido se compone de algunas unidades de producto.

**Esta relación generará una nueva tabla en la Base de datos. Puede incluir otros atributos como unidades, precio, etc.**

| Order id | Reference       |
|----------|-----------------|
| 12324    | 364834034843784 |

Esta tabla incluirá las CLAVES PRINCIPALES de las tablas de la relación.

# Big Data

# Big Data

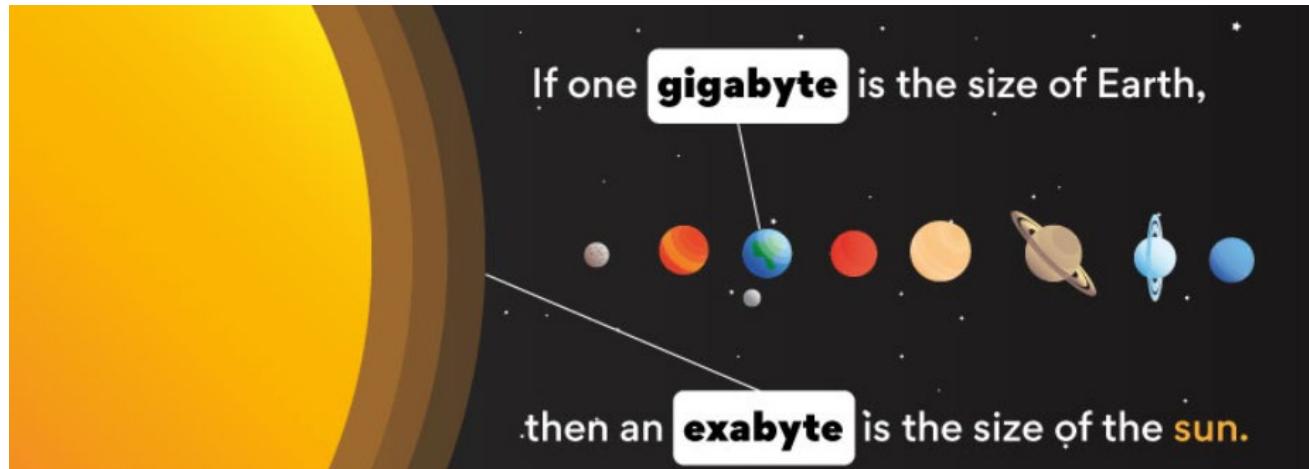


## Definición de "Big Data"

El término "Big Data" describe el gran volumen de datos, tanto estructurados como no estructurados, que se crean diariamente por la sociedad actual.

# Big Data

Desde la escritura cuneiforme, el sistema de escritura más antiguo conocido hasta la fecha, hasta los centros de datos modernos, la raza humana siempre ha recopilado información. Además, se predice que para 2030 nuestra civilización generará varios yottabytes de información por año..



El yottabyte es actualmente la mayor unidad reconocida de almacenamiento de datos para dispositivos y servicios en la nube.

# Leyes y ética

## Evolución de las leyes de datos en España

**1978:** Constitución Española. Art. 18.4 donde se garantiza el derecho de las personas al honor y la intimidad personal y familiar.

**1992:** LORTAD. Es la Ley Orgánica de Protección del Tratamiento Automatizado de los Datos de Carácter Personal (no vigente en la actualidad).

**1994:** Reglamento que desarrolla determinados aspectos de la LORTAD (este Reglamento sigue vigente a pesar de la derogación de la LORTAD).

**1995:** Directiva comunitaria relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos. La LOPD española se deriva de esta Directiva.

**1999:** Reglamento de Medidas de Seguridad (RMS). Especifica las medidas de seguridad técnicas y organizativas que se deben adoptar para los ficheros que contengan datos de carácter personal (11 de junio de 1999).

**1999: LOPD - Ley Orgánica de Protección de Datos de Carácter Personal** (adaptación de la antigua LORTAD a la Directiva Comunitaria de 1995). La ley sólo se aplica a los datos personales de las personas físicas, no de las personas jurídicas (empresas) (13 de diciembre de 1999).

## LOPD - Ley Orgánica de Protección de datos

La Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal (LOPD) fue una ley orgánica española que garantizaba y protegía el tratamiento de datos personales, las libertades públicas y los derechos humanos fundamentales, y especialmente del honor y la intimidad personal y familiar..

- Regular el tratamiento de los datos y ficheros, de carácter personal, independientemente del soporte en el que sean tratados.
- Los derechos de los ciudadanos sobre ellos y las obligaciones de quienes los crean o tratan.

## LOPD en la Unión Europea

El GRDP (Reglamento General de Protección de Datos) es una regulación en la legislación de la Unión Europea (UE) sobre protección de datos y privacidad en el Espacio Económico Europeo (EEE). Esta ley es obligatoria desde el 25 de mayo de 2018 en todos los países miembros de la Unión Europea.

La LOPD, como ley nacional (España), se integró en el RGPD, que unificó toda la normativa europea de protección de datos bajo un mismo paraguas legal. Para que todos los ciudadanos europeos tengan los mismos derechos y garantías con respecto a la información personal.

**LOPD == GRDP**

## LOPD en la Unión Europea

La LOPD-GDD (Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales) es la nueva ley nacional integrada en el GRPD, que unificó toda la normativa europea de protección de datos bajo un mismo paraguas legal. Esta ley amplía la normativa definida en el RGPD.

# Cómo implantar la nueva LOPD-GDD en una empresa?

## LOPD en la Unión Europea

Cualquier tipo de empresa o negocio que trate datos sensibles de terceros, deberá cumplir con todos y cada uno de los requisitos establecidos en la nueva normativa de la Ley de Protección de Datos Personales y Garantías de los Derechos Digitales. La LOPD-GDD se aplicará cuando se administren los siguientes tratamientos:

- Tratamiento de datos de empresarios individuales y profesionales liberales.
- Operaciones comerciales.
- Uso de sistemas de vigilancia.
- Sistemas de exclusión publicitaria.
- Canales de comunicación y denuncias.
- Sistemas de información crediticia.

## LOPD en la Unión Europea

Hay algunos puntos importantes en la nueva ley:

**Protección de menores:** El consentimiento de un menor sólo será válido cuando sea mayor de catorce años, siendo necesaria la autorización del padre, madre o tutor si no lo fuere.

**Control de datos personales:** Para evitar el uso de datos personales para uso comercial sin consentimiento previo, la LOPD-GDD establece que el control de los datos personales recae directamente en el usuario, requiriendo siempre su consentimiento para utilizarlos.

**Privacidad de los empleados:** Está prohibido realizar grabaciones en las zonas destinadas al descanso de los trabajadores, aseos y otros lugares destinados al ocio.

## LOPD en la Unión Europea

Hay algunos puntos importantes en la nueva ley:

**Derecho al olvido:** establece el derecho a suprimir datos en redes sociales y otros servicios equivalentes.

**Datos de personas fallecidas:** En caso de fallecimiento, cualquier familiar vinculado a la persona fallecida podrá solicitar el acceso, rectificación o supresión de los datos compartidos.

**Información clara sobre el uso de los datos:** Las empresas deben informar a los usuarios de forma clara, sencilla y concisa sobre el posible uso de los datos personales que se les han facilitado.

**Las empresas podrían ser multadas con hasta 20 millones de euros.**

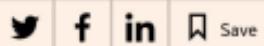
# ¿Qué pasa con la ética?

The Big Read Artificial intelligence

+ Add to myFT

## Insurance: Robots learn the business of covering risk

Artificial intelligence could revolutionise the industry but may also allow clients to calculate if they need protection



Oliver Ralph MAY 16, 2017

24

 Stanford MEDICINE News Center

Email

Tweet

Researchers say use of artificial intelligence in medicine raises ethical questions

In a perspective piece, Stanford researchers discuss the ethical implications of using machine-learning tools in making health care decisions for patients.

## After Uber, Tesla incidents, can artificial **intelligence** be trusted?

Apr 9, 2018 | News Stories

The reliability of self-driving cars and other forms of artificial intelligence is one of several factors that affect humans' trust in AI, **machine learning** and other technological advances, write two Missouri University of Science and Technology researchers in a recent journal article. "Trust is the cornerstone of ...

# Ética

La ética, también llamada filosofía moral, la disciplina que se ocupa de lo que es moralmente bueno y malo y moralmente correcto e incorrecto. El término también se aplica a cualquier sistema o teoría de valores o principios morales..

## ¿Por qué necesitamos ética cuando estamos creando software?

- Los humanos tenemos sesgos que incluimos en la información que creamos y analizamos.
- Durante el diseño y desarrollo de software y datos aportamos conocimiento inconsciente, debemos preguntarnos si hemos tenido en cuenta suficientes ejemplos.
- Ideologías, expresiones, información técnica validada, luz natural/artificial, casos atípicos, etc..

# Ética

<https://www.moralmachine.net/>



## Option 1

In this case, the self-driving car with a sudden brake failure will swerve and drive through a crosswalk in the other lane. This will result in the death of an elderly woman, two athletes and a child.

Please note that affected pedestrians are complying with the law when crossing with the green signal

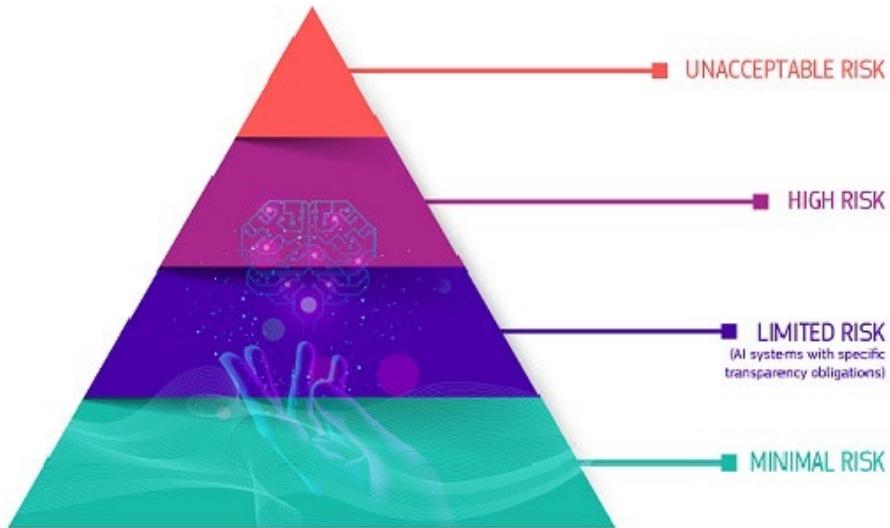
## Option 2

In this case, the self-driving car with sudden brake failure will continue forward and crash into a concrete barrier. This will result in the death of a child, a pregnant woman (passengers) and a dog."

<https://www.eeworldonline.com/this-mit-game-lets-you-choose-who-lives-and-dies-in-a-self-driving-car-wreck/>

# Cómo estamos tratando de crear ¿Software o datos éticamente?

## Nuevo marco regulatorio sobre IA (Unión Europea).



- **Riesgo inaceptable:** un conjunto muy limitado de usos especialmente perjudiciales de la IA que contravienen los valores de la UE porque violan los derechos fundamentales. Puntuación social para los gobiernos (Esto está sucediendo en China).
- **Alto riesgo:** Un número limitado de sistemas de IA definidos en la propuesta que crean un impacto adverso en la seguridad de las personas o sus derechos fundamentales (protegidos por la Carta de los Derechos Fundamentales de la UE) se consideran de alto riesgo.
  - Infraestructura.
  - Educación.
  - Seguridad.
  - Servicios públicos.
  - Controles de inmigración o de líneas fronterizas.
- **Riesgo limitado:** Para determinados sistemas de IA se imponen requisitos específicos de transparencia, por ejemplo, cuando existe un riesgo claro de manipulación (por ejemplo, mediante el uso de chatbots). Los usuarios deben ser conscientes de que están interactuando con una máquina.
- **Riesgo mínimo:** Todos los demás sistemas de IA pueden desarrollarse y utilizarse sujetos a la legislación vigente sin obligaciones legales adicionales.
- La gran mayoría de los sistemas de IA utilizados actualmente en la UE entran en esta categoría.

Hay reglas en otros países y estados.

- **GDPR:** Article 22 empowers individuals with the **right to demand an explanation of how an automated system made a decision** that affects them.
- **Algorithmic Accountability Act 2019:** Requires companies to **provide an assessment of the risks** posed by the automated decision system to the **privacy or security** and the risks that contribute to **inaccurate, unfair, biased, or discriminatory decisions** impacting consumers
- **California Consumer Privacy Act:** Requires companies to **rethink their approach to capturing, storing, and sharing personal data** to align with the new requirements by January 1, 2020.
- **Washington Bill 1655:** Establishes guidelines for the use of automated decision systems to protect consumers, improve transparency, and create more market predictability.
- **Massachusetts Bill H.2701:** Establishes a commission on **automated decision-making, transparency, fairness, and individual rights**.
- **Illinois House Bill 3415:** States predictive data analytics determining creditworthiness or hiring decisions **may not include information that correlates** with the applicant race or zip code.