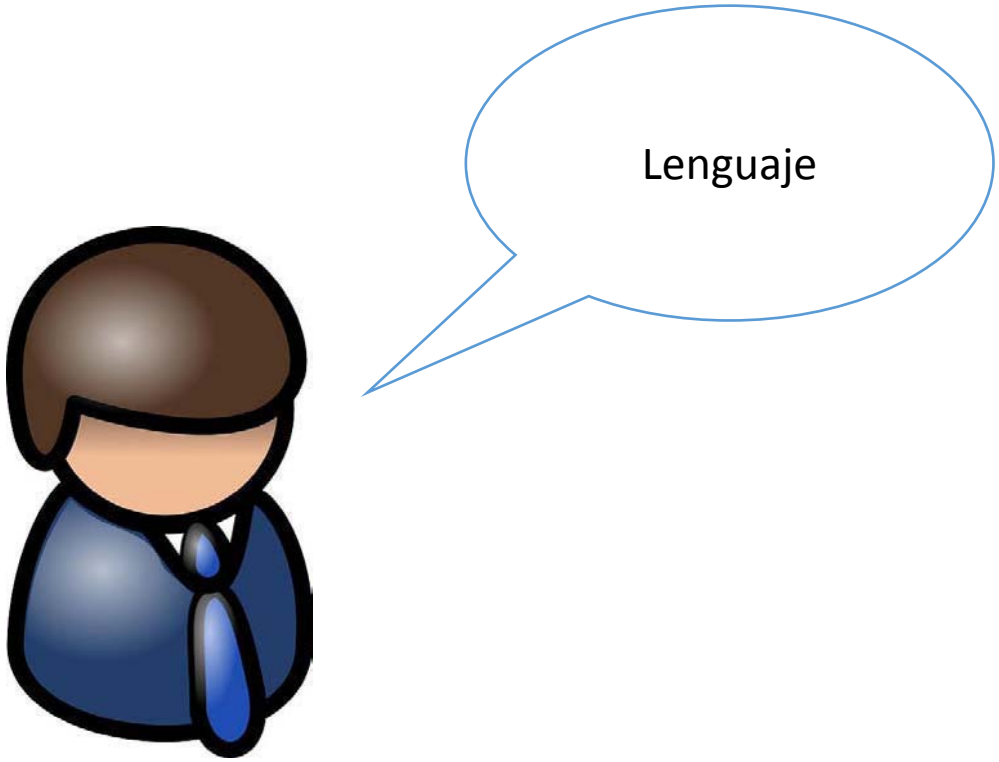
Tema 1: Introducción al curso

http://pixabay.com/es/iphone-tel%C3%A9fono-celular-smartphone-37856/
http://pixabay.com/es/tablet-ipad-pantalla-internet-184888/
http://pixabay.com/es/ordenador-port%C3%A1til-port%C3%A1til-m%C3%B3viles-154091/

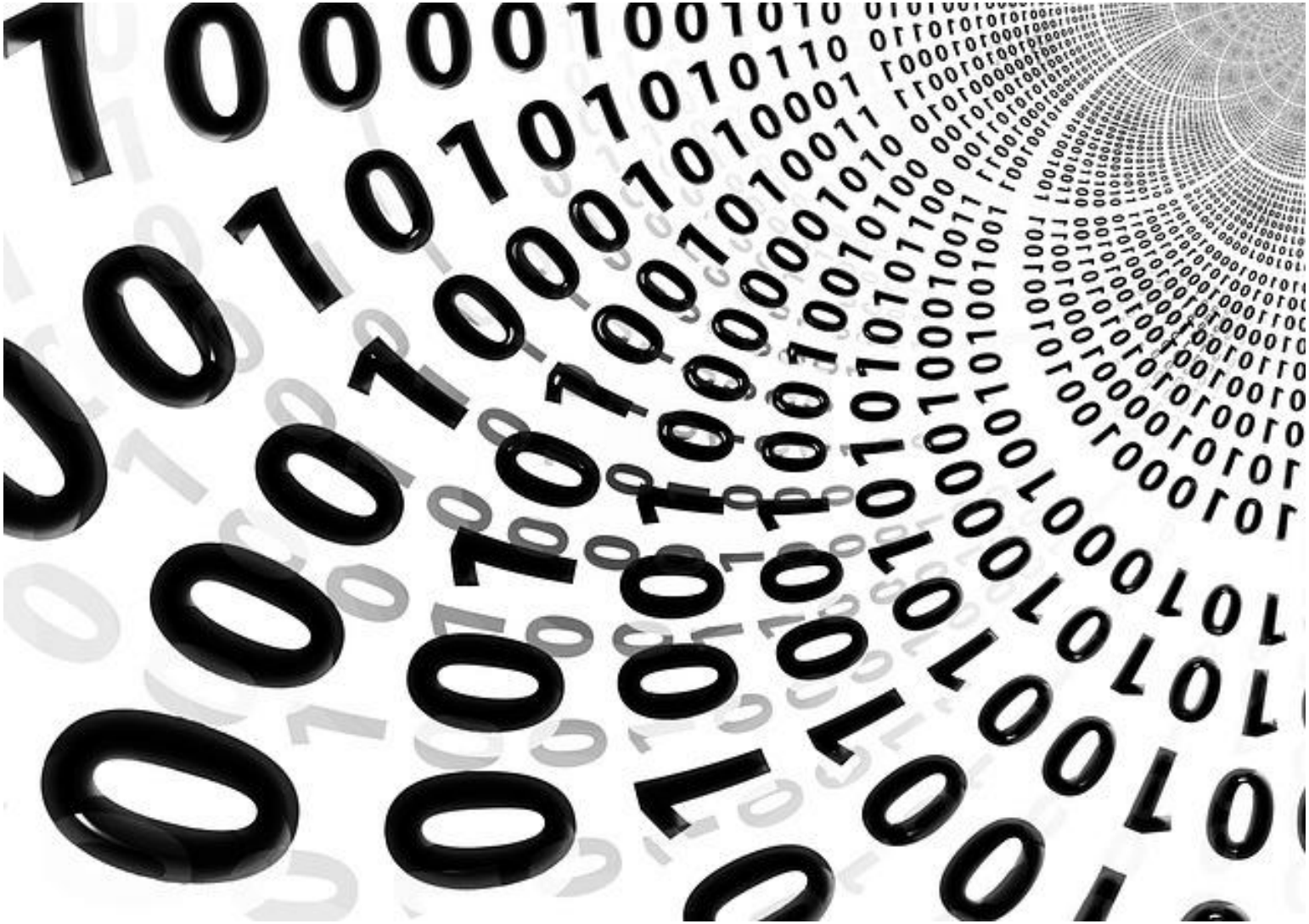
Información

http://pixabay.com/es/iphone-tel%C3%A9fono-celular-smartphone-37856/



http://pixabay.com/es/hombre-persona-corbata-traje-23850/

http://pixabay.com/es/ordenador-port%C3%A1til-port%C3%A1til-m%C3%B3viles-154091/



http://pixabay.com/es/c%C3%B3digo-binario-binaria-507785/

En este curso

- Codificación:
  - Enteros positivos
  - Enteros negativos
  - Reales positivos y negativos
  - Caracteres alfanuméricos

http://i.ytimg.com/vi/9bZkp7q19f0/maxresdefault.jpg

Ejemplo de la importancia de la codificación

http://i.ytimg.com/vi/9bZkp7q19f0/maxresdefault.jpg

Organización del curso

- Semana 1:
  - Introducción a los sistemas de numeración
- Semana 2:
  - Sistema de numeración binario
- Semana 3:
  - Sistema de numeración hexadecimal
- Semana 4:
  - Codificación de enteros positivos
- Semana 5:
  - Codificación de enteros negativos
- Semana 6:
  - Codificación de números reales
- Semana 7:
  - Codificación de caracteres alfanuméricos
- Semana 8:
  - Examen final

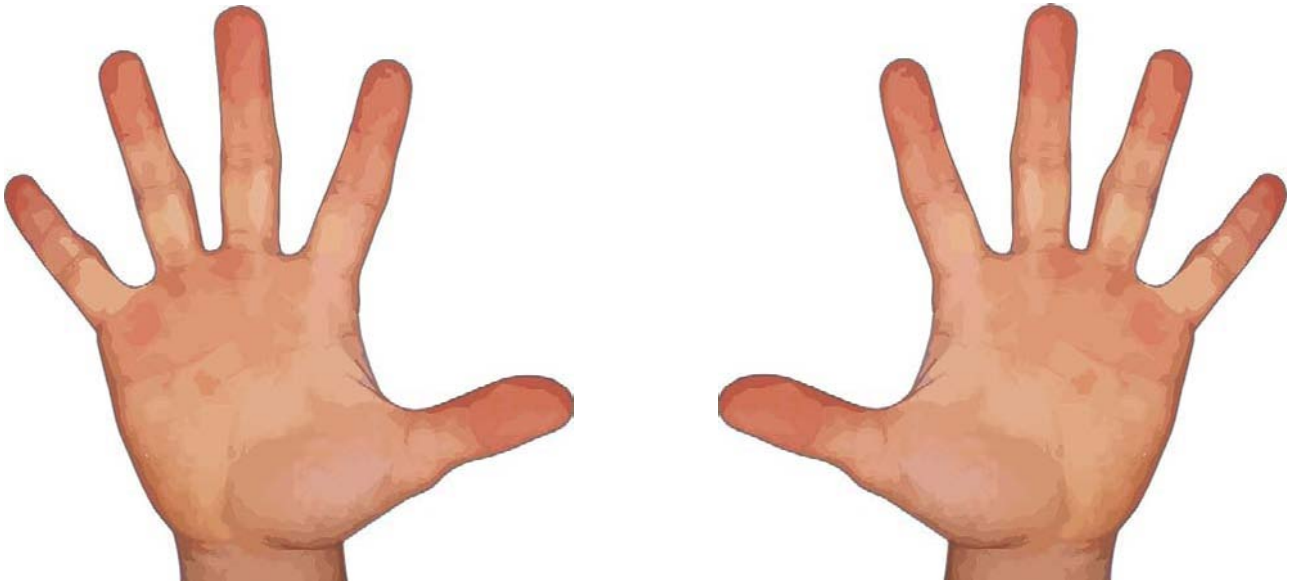
Evaluación del curso

- Cada semana habrá una prueba de conocimientos:
  - Se puede repetir **infinitas** veces.
  - Nota final, la máxima.
  - En total 7 pruebas

- En la semana 8: Examen final
  - Se podrá repetir un número **finito** de veces.

- Nota del curso:
  - Examen final * 0.5 + media pruebas de cada semana * 0.5

El sistema de numeración decimal

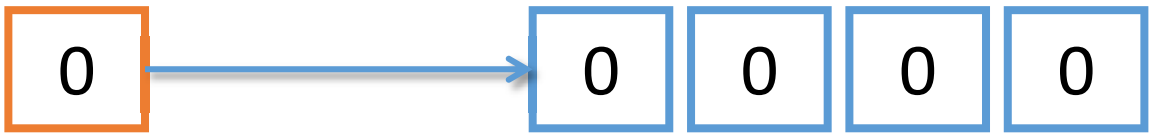


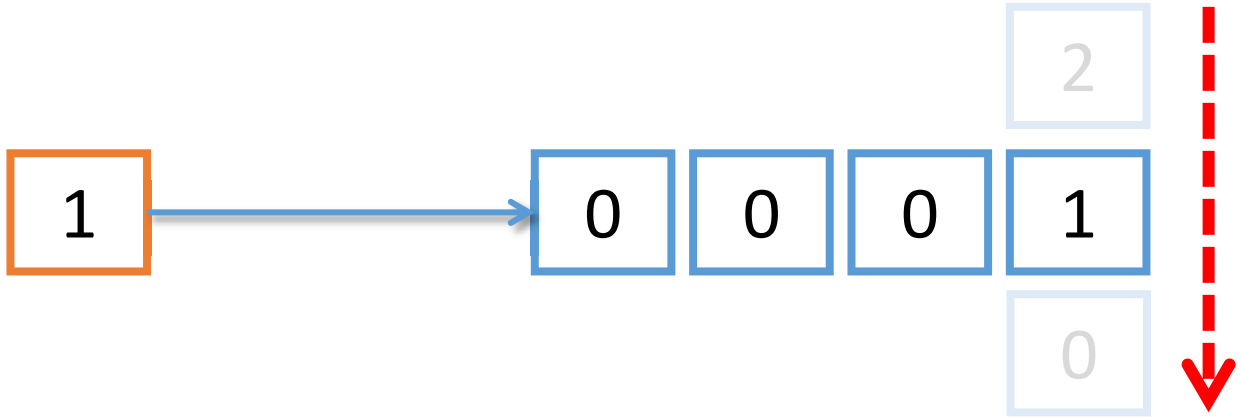
http://pixabay.com/es/dedo-mano-pulgar-dedo-%C3%ADndice-160597/

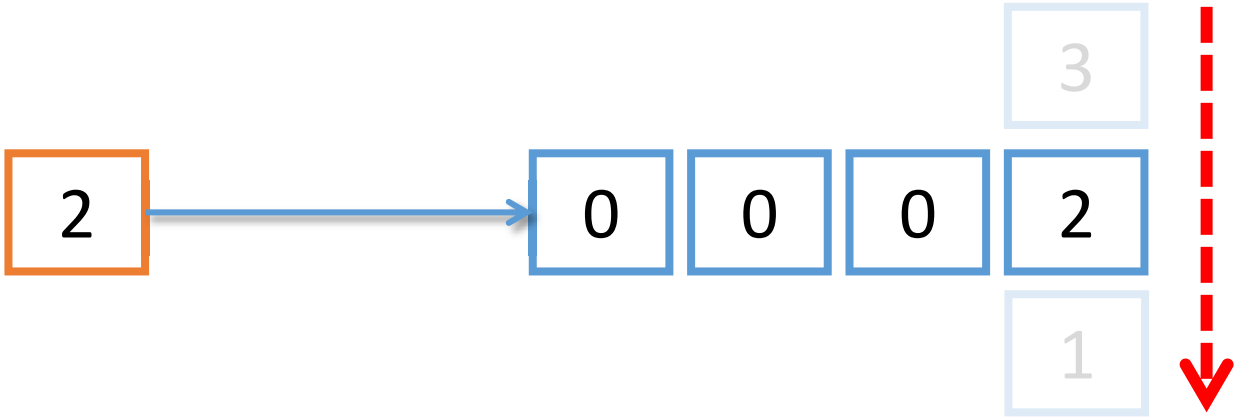
- **Base**:
  - Número de símbolos usados por el sistema de numeración.

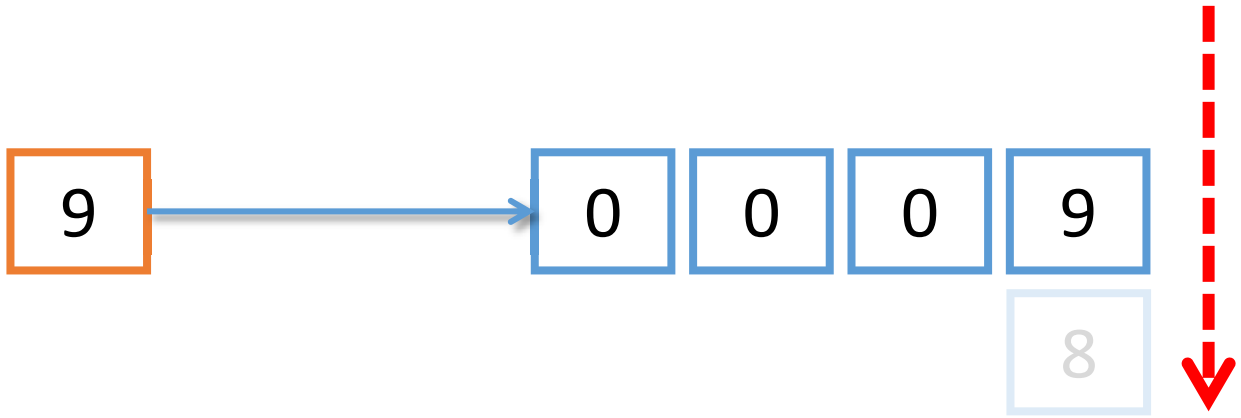
- El sistema decimal tiene base 10:
  - Símbolos: 0,1,2,3,4,5,6,7,8,9

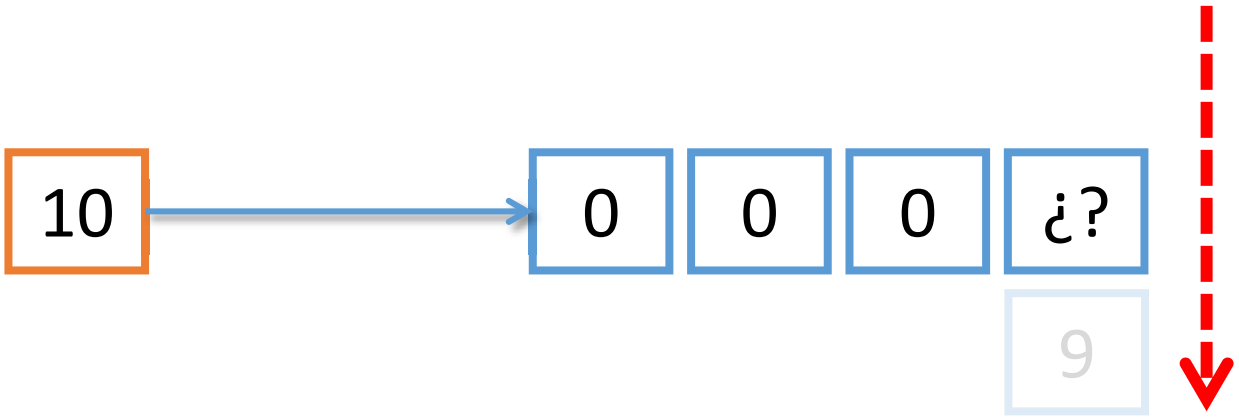
- ¿Cómo funciona el sistema de numeración decimal?
- Supongamos N=4
  - N es el número de dígitos que podemos usar

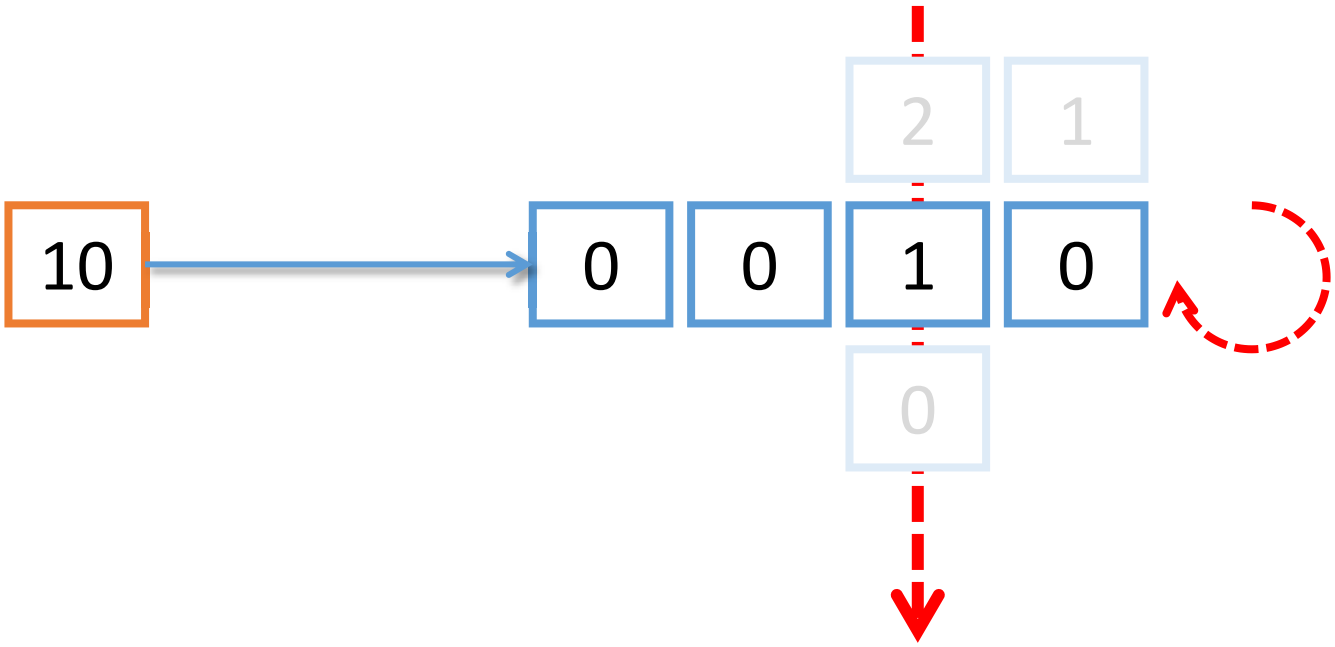


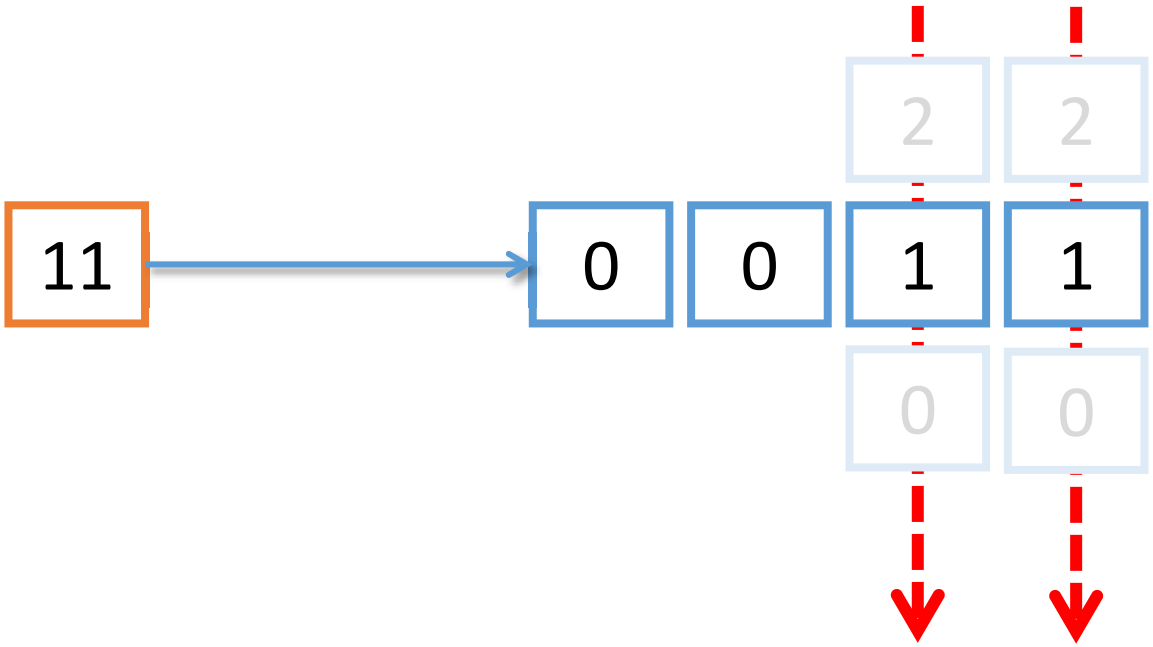

2
1
0
0
0
1
0

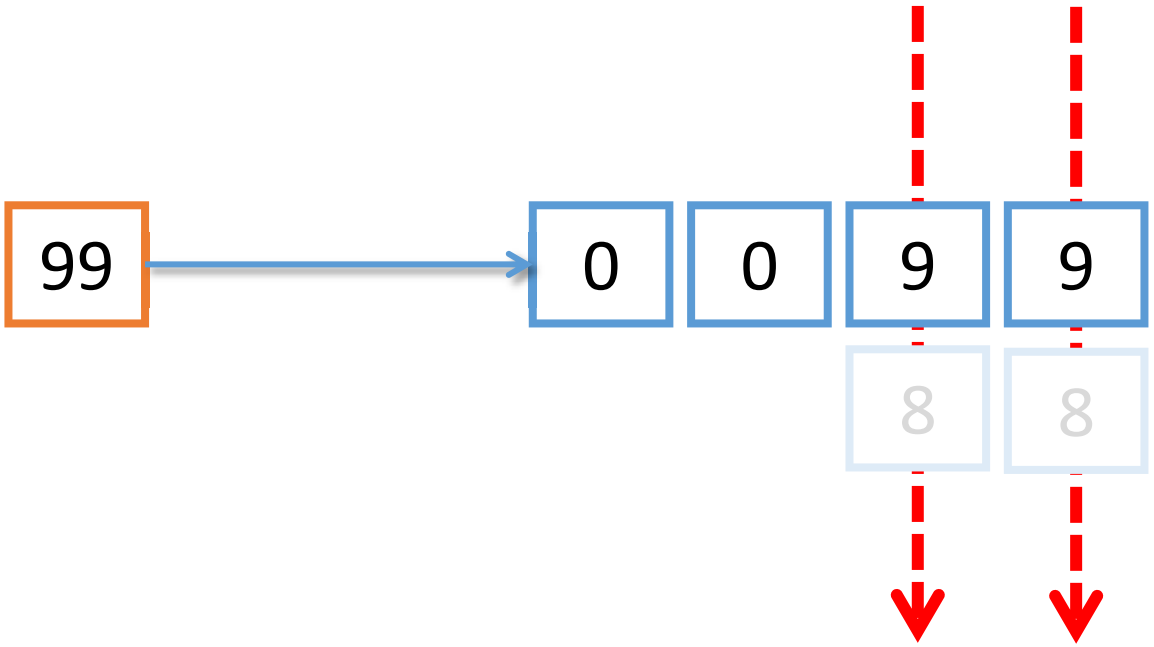

3
2
0
0
0
2
1

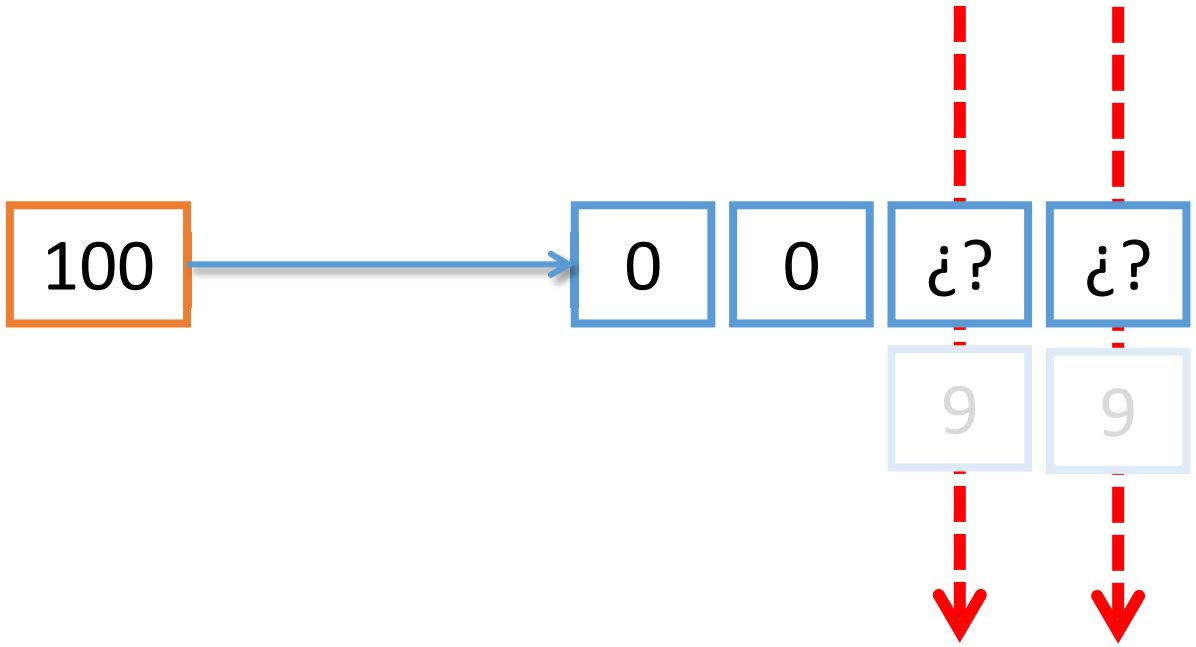

9
0
0
0
9
8

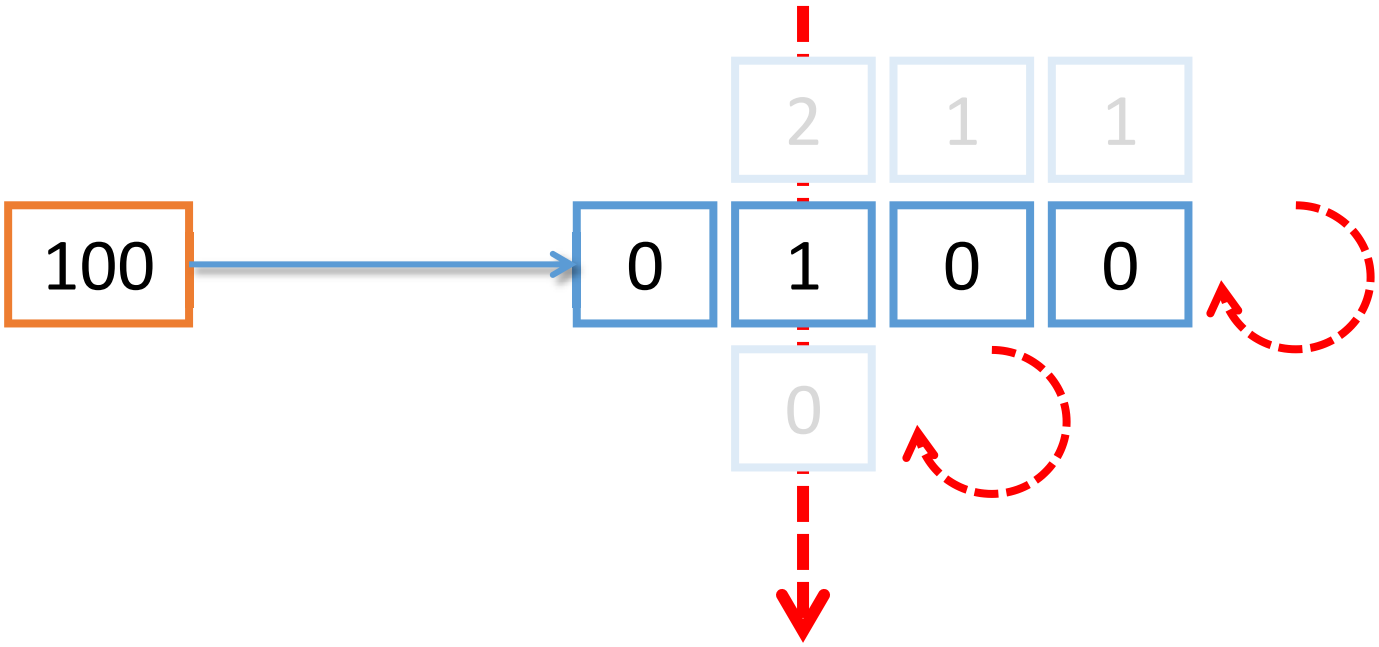

10
0
0
0
¿?
9

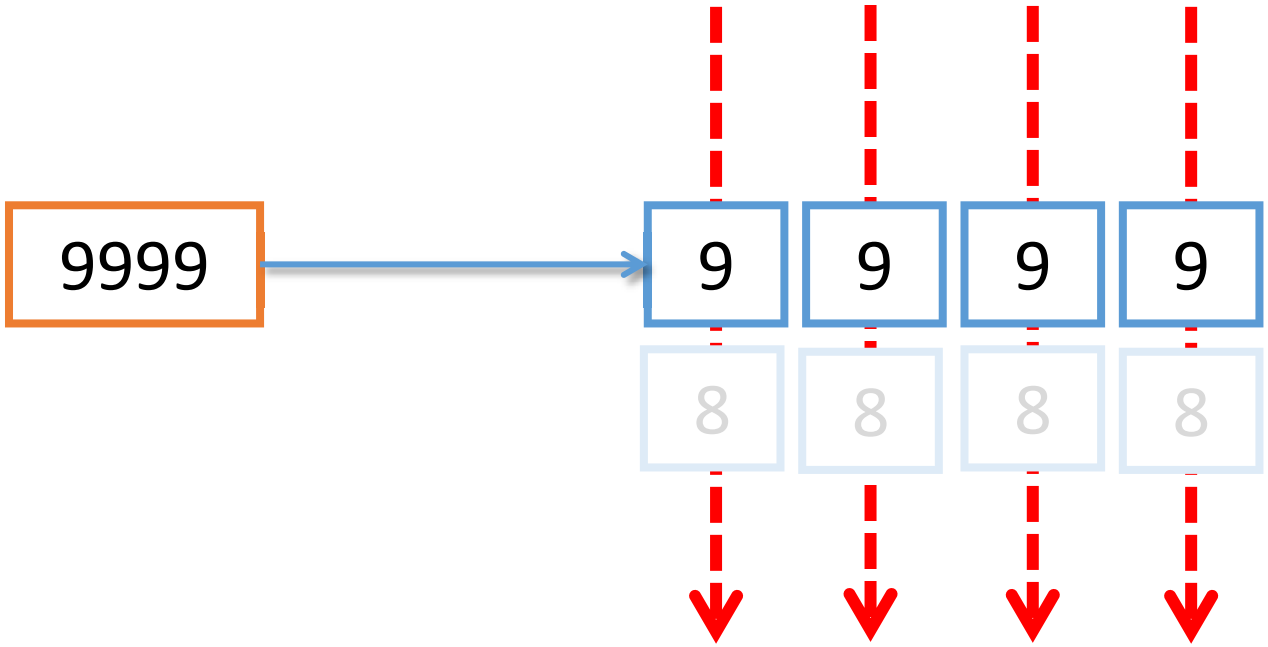

2
1
10
0
0
1
0
0

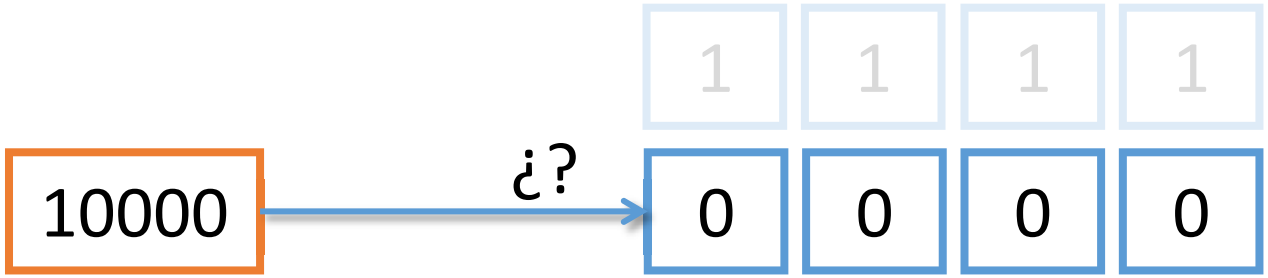

11
0
0
1
1
2
2
0
0


99
0
0
9
9
8
8


100
0
0
¿?
¿?
9
9


100
0
1
0
0
2
1
1
0


9999
9
9
9
9
8
8
8
8


1
1
1
1
10000
¿?
0
0
0
0

- Con N dígitos, es posible representar el rango:

$$[\ 0,\ 10^{N} - 1]$$

- Con N=4 dígitos:

$$[0000,\ 9999]$$

Los sistemas de numeración posicionales

- En un sistema de numeración posicional:
  - Cada dígito posee un valor que depende de su posición relativa y del valor del símbolo.
  - Existen un número finito de símbolos = base

- El sistema de numeración Romano:
  - No es posicional

  - Cuando surge la necesidad de representar nuevas cantidades es necesario añadir más símbolos:
    - Número de símbolos no es fijo
  - Dificulta las matemáticas

- Conversión de cualquier base (x_b) a decimal (x_{10}): números enteros

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- $x_b[p]$ -> valor decimal del símbolo situado en la posición p-ésima

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

754_{10}

- b=10
- N=3
- p=2,1,0

$x_b[2] = 7$
$x_b[1] = 5$
$x_b[0] = 4$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

754_{10} =

$= x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

754_{10} =

$= x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0$

$= 7*10^2 + 5*10^1 + 4*10^0$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

754_{10} =

$= x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0$

$= 7*10^2 \quad + 5*10^1 \quad + 4*10^0$

$= 700 \quad + 50 \quad + 4$

- Conversión de cualquier base (x_b) a decimal (x_{10}): números reales

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

Ejemplo

$305{,}45_{10}$

- b = 10
- $N^+ = 3$
- $N^- = 2$
- p = 2,1,0
- q = -1,-2

$x_b[2] = 3$ $\quad$ $x_b[-1] = 4$

$x_b[1] = 0$ $\quad$ $x_b[-2] = 5$

$x_b[0] = 5$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

Ejemplo

$305{,}45_{10}$ =

$= x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0 +$

$x_b[-1]*10^{-1} + x_b[-2]*10^{-2}$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

Ejemplo

$305{,}45_{10}$ =

$= x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0 +$

$x_b[-1]*10^{-1} + x_b[-2]*10^{-2}$

$= 3*10^2 + 0*10^1 + 5*10^0 +$

$4*10^{-1} + 5*10^{-2}$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

Ejemplo

$305{,}45_{10}$ =

$= x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0 +$

$x_b[-1]*10^{-1} + x_b[-2]*10^{-2}$

$= 3*10^2 + 0*10^1 + 5*10^0 +$

$4*10^{-1} + 5*10^{-2}$

$= 300 + 0 + 5 +$

$0{,}4 + 0{,}05$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

Ejemplo

$3876{,}213_{10}$

- b = 10
- $N^+ = 4$
- $N^- = 3$
- p = 3,2,1,0
- q = -1,-2,-3

$x_b[3] = 3$ $x_b[-1] = 2$

$x_b[2] = 8$ $x_b[-2] = 1$

$x_b[1] = 7$ $x_b[-3] = 3$

$x_b[0] = 6$

$$x_{10} = \sum_{p=N^{+}-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^{-}} x_b[q] * b^q$$

Ejemplo

$3876{,}213_{10}$ =

= $x_b[3]*10^3 + x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0 +$

$x_b[-1]*10^{-1} + x_b[-2]*10^{-2} + x_b[-3]*10^{-3}$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

Ejemplo

$3876{,}213_{10}$ =

= $x_b[3]*10^3 + x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0 +$

$x_b[-1]*10^{-1} + x_b[-2]*10^{-2} + x_b[-3]*10^{-3}$

= $3*10^3 + 8*10^2 + 7*10^1 + 6*10^0 +$

$2*10^{-1} + 1*10^{-2} + 3*10^{-3}$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

Ejemplo

$3876{,}213_{10}$ =

$= x_b[3]*10^3 + x_b[2]*10^2 + x_b[1]*10^1 + x_b[0]*10^0 +$
$x_b[-1]*10^{-1} + x_b[-2]*10^{-2} + x_b[-3]*10^{-3}$

$= 3*10^3 + 8*10^2 + 7*10^1 + 6*10^0 +$
$2*10^{-1} + 1*10^{-2} + 3*10^{-3}$

= 3000 + 800 + 70 + 6 +
0,2 + 0,01 + 0,003

- Rango de posibles valores dada una base b y número de dígitos N:

$$[0, b^N - 1]$$

Ejemplos:

- Con b=3, N=5:

$[0,b^N-1] = [0,3^5-1] = [0,242]$

Ejemplos:

- Con b=7, N=4:

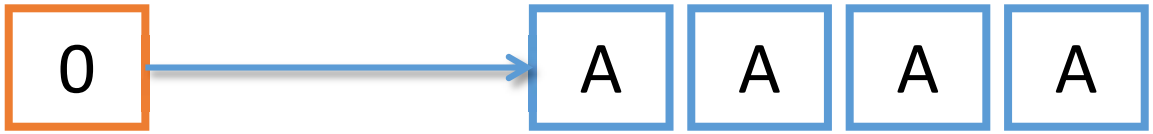
$[0,b^N-1] = [0,7^4-1] = [0,2400]$

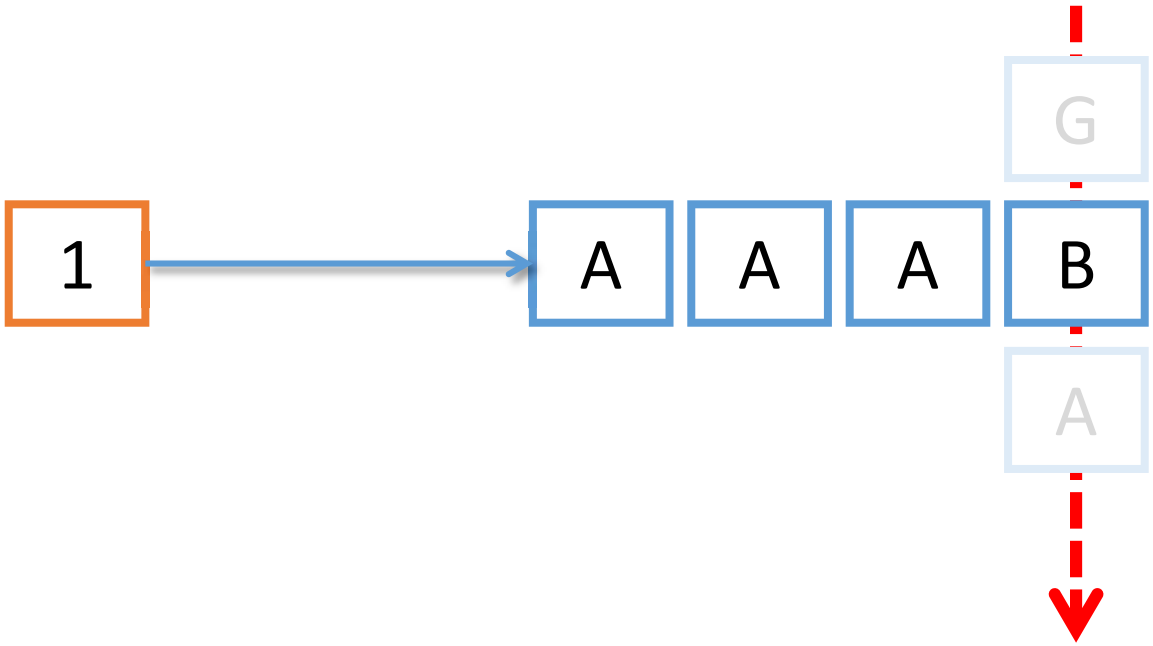
- Bases más pequeñas necesitan más dígitos para representar los mismos números:

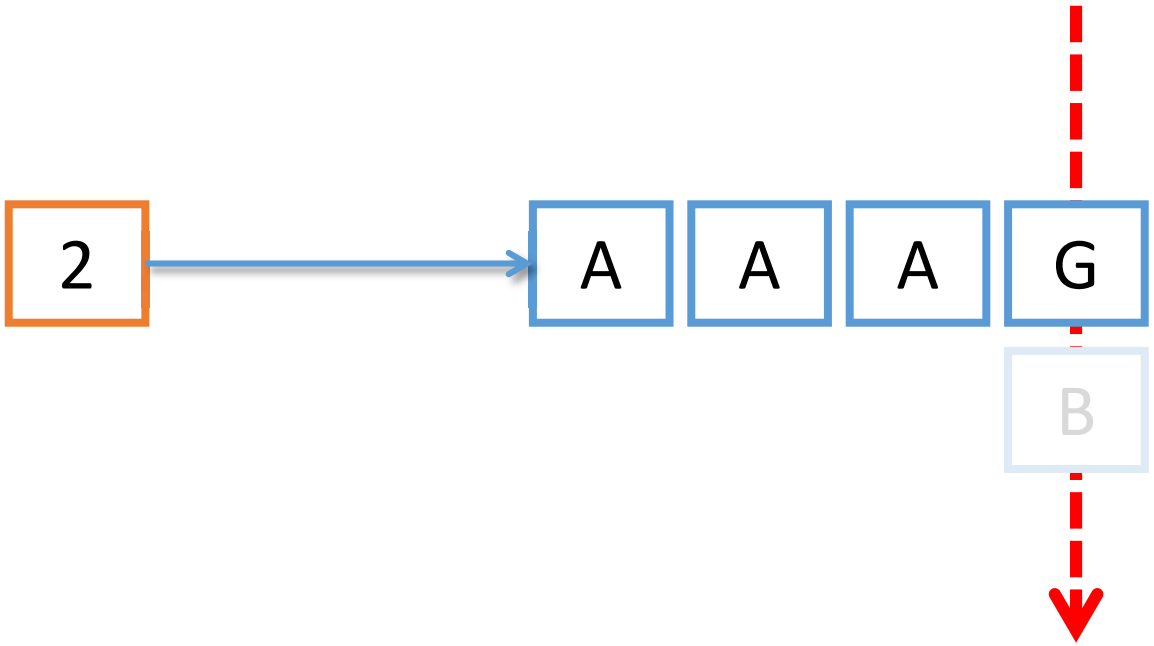
- Ejemplo, para representar el 70:
  - b=10, hacen falta 2 dígitos
  - b=8, hacen falta 3 dígitos
  - b=2, hacen falta 7 dígitos

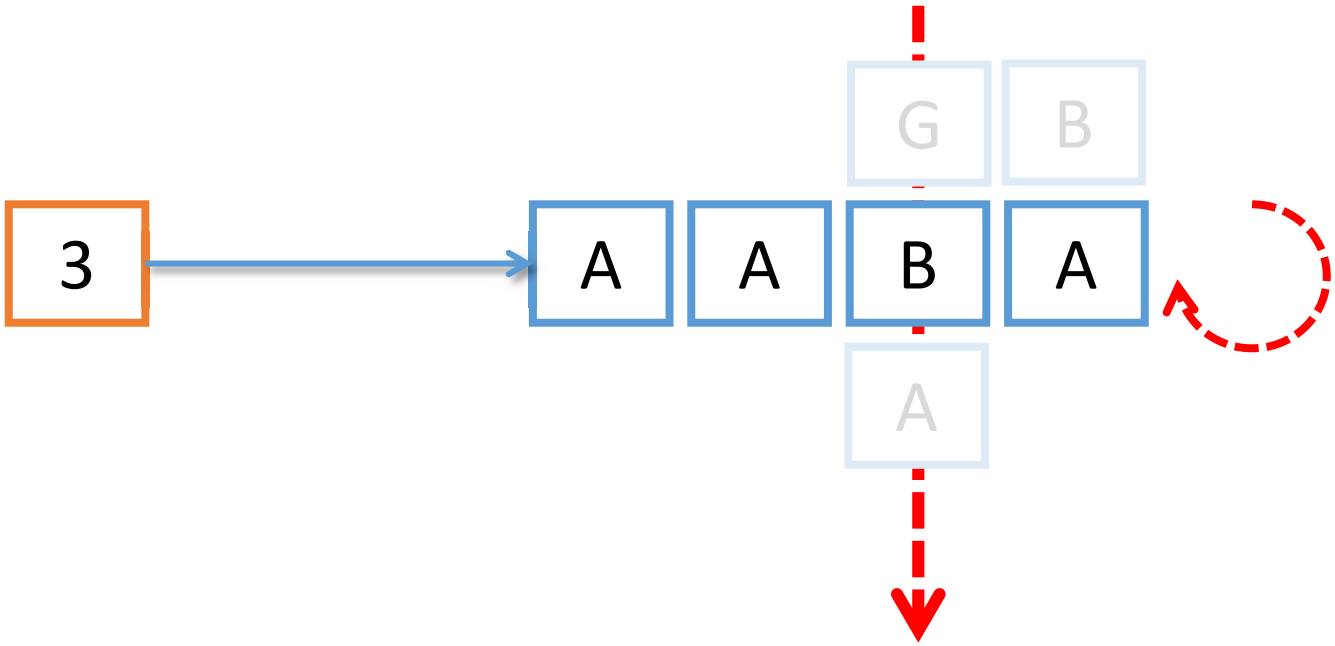
Tri: un sistema en base 3 para practicar

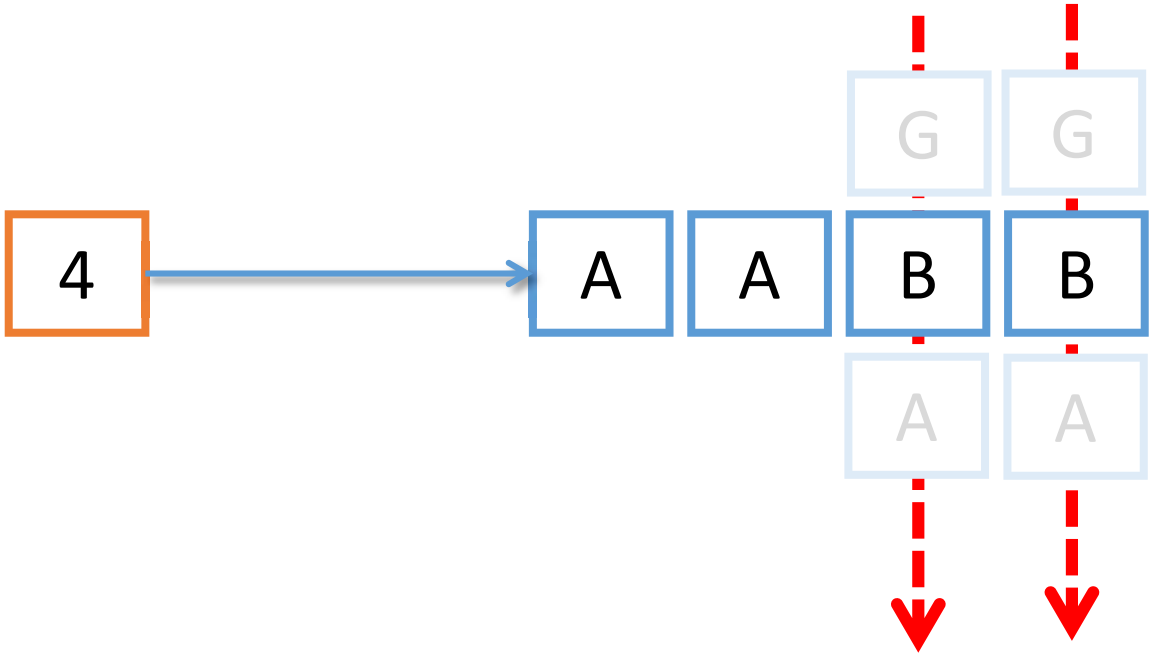
- Sistema de base 3 (b=3):
  - A -> 0
  - B- > 1
  - G -> 2

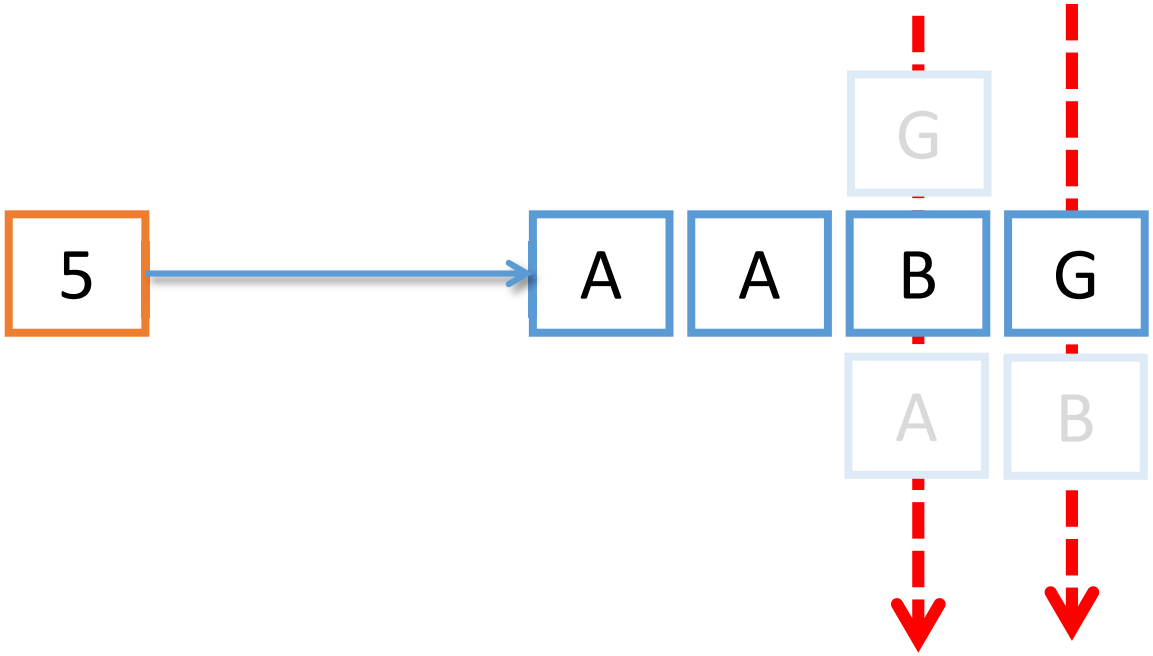

0
A
A
A
A

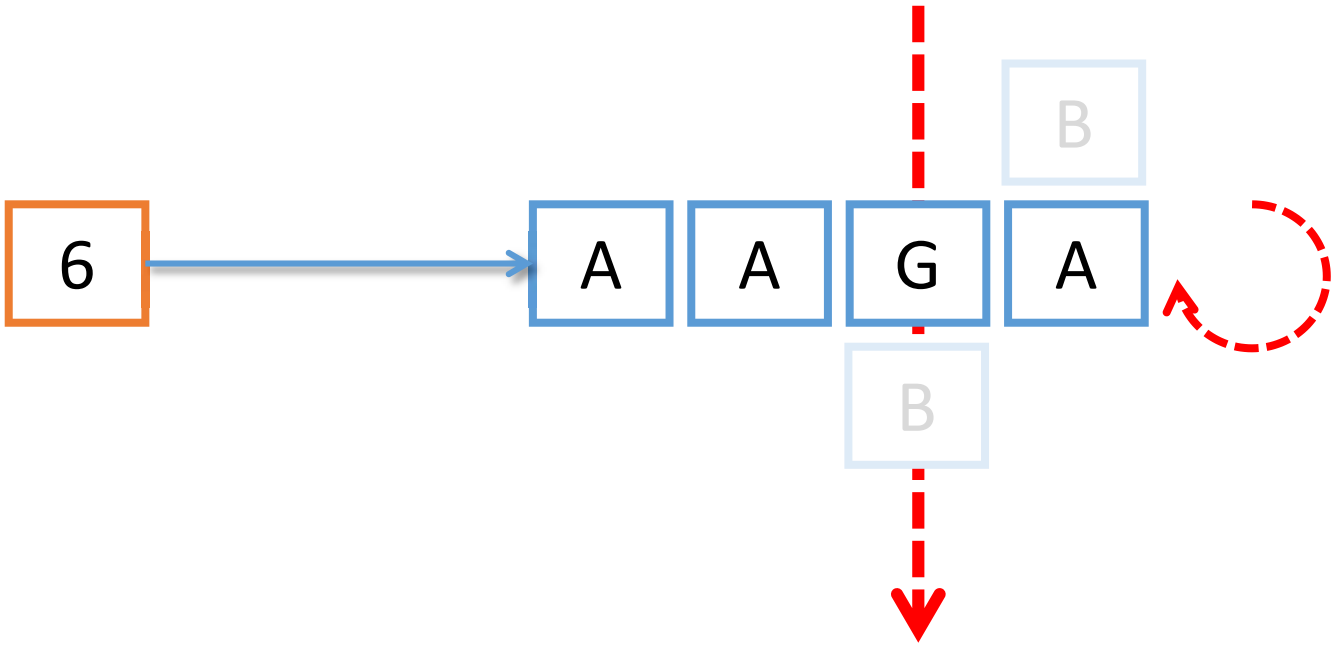

1
A
A
A
B
G
A


2
A
A
A
G
B


3
A
A
B
A
G
B
A


4
A
A
B
B
G
G
A
A


5
A
A
B
G
G
A
B


6
A
A
G
A
B
B

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

A -> 0
B -> 1
G -> 2

$AAGA_3$

- b = 3
- N = 4
- p=3,2,1,0

$X_b[3] = 0$ (el valor decimal del símbolo A es 0)
$X_b[2] = 0$
$X_b[1] = 2$
$X_b[0] = 0$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

A -> 0
B -> 1
G -> 2

- $AAGA_3$ =

  $= x_b[3]*3^3 + x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

A -> 0
B -> 1
G -> 2

- $AAGA_3$ =

  = $x_b[3]*3^3 + x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0$

  = $0*3^3 + 0*3^2 + 2*3^1 + 0*3^0$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

A -> 0
B -> 1
G -> 2

- $AAGA_3$ =

$= x_b[3]*3^3 + x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0$

$= 0*3^3 + 0*3^2 + 2*3^1 + 0*3^0$

$= 0*27 + 0*9 + 2*3 + 0*3$

$= 0 + 0 + 6 + 0$

$= 6_{10}$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

A -> 0
B -> 1
G -> 2

BGB_3

- b = 3
- N = 3
- p=2,1,0

$X_b[2] = 1$
$X_b[1] = 2$
$X_b[0] = 1$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

A -> 0
B -> 1
G -> 2

- BGB_3 =

  $= x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

A -> 0
B -> 1
G -> 2

- BGB_3 =

  $= x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0$

  $= 1*3^2 + 2*3^1 + 1*3^0$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

A -> 0
B -> 1
G -> 2

- BGB_3 =
  - $= x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0$
  - $= 1*3^2 + 2*3^1 + 1*3^0$
  - $= 9 + 6 + 1$
  - $= 16$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

BBG,AB_3

A -> 0
B -> 1
G -> 2

- b = 3
- $N^+ = 3$
- $N^- = 2$
- p = 2,1,0
- q = -1,-2

$x_b[2] = 1$ $x_b[-1] = 0$
$x_b[1] = 1$ $x_b[-2] = 1$
$x_b[0] = 2$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

A -> 0
B -> 1
G -> 2

- BBG,AB_3 =

  $= x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0 +$

  $x_b[-1]*3^{-1} + x_b[-2]*3^{-2}$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

A -> 0
B -> 1
G -> 2

- BBG,AB_3 =

  $= x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0 +$

  $x_b[-1]*3^{-1} + x_b[-2]*3^{-2}$

  $= 1*3^2 + 1*3^1 + 2*3^0 +$

  $0*3^{-1} + 1*3^{-2}$

$$x_{10} = \sum_{p=N^{+}-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^{-}} x_b[q] * b^q$$

A -> 0
B -> 1
G -> 2

- BBG,AB_3 =

$= x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0 +$
$x_b[-1]*3^{-1} + x_b[-2]*3^{-2}$
$= 1*3^2 + 1*3^1 + 2*3^0 +$
$0*3^{-1} + 1*3^{-2}$
$= 1*9 + 1*3 + 2*1 +$
$0*1/3 + 1*1/9$

$$x_{10} = \sum_{p=N^{+}-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^{-}} x_b[q] * b^q$$

- BBG,AB_3 =

  $= x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0 +$
  $x_b[-1]*3^{-1} + x_b[-2]*3^{-2}$

  $= 1*3^2 + 1*3^1 + 2*3^0 +$
  $0*3^{-1} + 1*3^{-2}$

  $= 1*9 + 1*3 + 2*1 +$
  $0*1/3 + 1*1/9$

  $= 9 + 3 + 2 +$
  $0 + 1/9$

  $= 14.11111\ldots_{10}$

A -> 0
B -> 1
G -> 2

$$x_{10} = \sum_{p=N^{+}-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^{-}} x_b[q] * b^q$$

- BBG,AB_3 =

  $= x_b[2]*3^2 + x_b[1]*3^1 + x_b[0]*3^0 +$
  $x_b[-1]*3^{-1} + x_b[-2]*3^{-2}$

  $= 1*3^2 + 1*3^1 + 2*3^0 +$
  $0*3^{-1} + 1*3^{-2}$

  $= 1*9 + 1*3 + 2*1 +$
  $0*1/3 + 1*1/9$

  $= 9 + 3 + 2 +$
  $0 + 1/9$

  $= 14.11111\ldots_{10}$

  $= 14.\overset{\frown}{1}_{10}$

A -> 0
B -> 1
G -> 2

Bits y Bytes

Bits y bytes

- Bits: elemento capaz de almacenar un único dígito en binario.

- Byte: 8 bits.

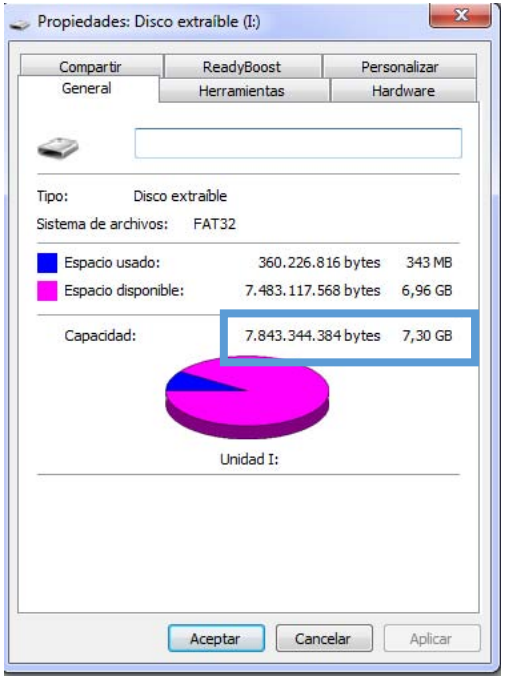
Bits y bytes

Nombre	Prefijo	Número bytes
kilobyte	KB	$10^3 = 1000$
Megabyte	MB	$10^6 = 1000000$
Gigabyte	GB	$10^9 = 1000000000$
Terabyte	TB	$10^{12} = 1000000000000$

Bits y bytes

Nombre	Prefijo	Número bytes
kibibyte	KiB	$2^{10} = 1024$
Mebibyte	MiB	$2^{20} = 1048576$
Gibibyte	GiB	$2^{30} = 1073741824$
Tebibyte	TiB	$2^{40} = 1099511627776$

Bits y bytes



- El fabricante: 8GB = $8*10^9$ bytes

  En realidad un poco menos: 7.843.344.384

- El Sistema operativo: ¿7.3GB?

Bits y bytes

- El sistema operativo no sigue las recomendaciones del SI e interpreta **erróneamente** que 8GB son $8*2^{30}$ bytes.
  - Es decir interpreta GB como GiB.
- ¿7.3GB?
  - 7.3 es aproximadamente $8*((10^9)/(2^{30}))$

Bits y bytes

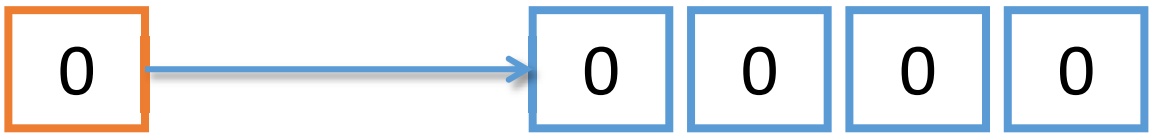
- Capacidad de almacenamiento:
  - En bytes y sus múltiplos
- Velocidad de transmisión:
  - En bits y sus múltiplos

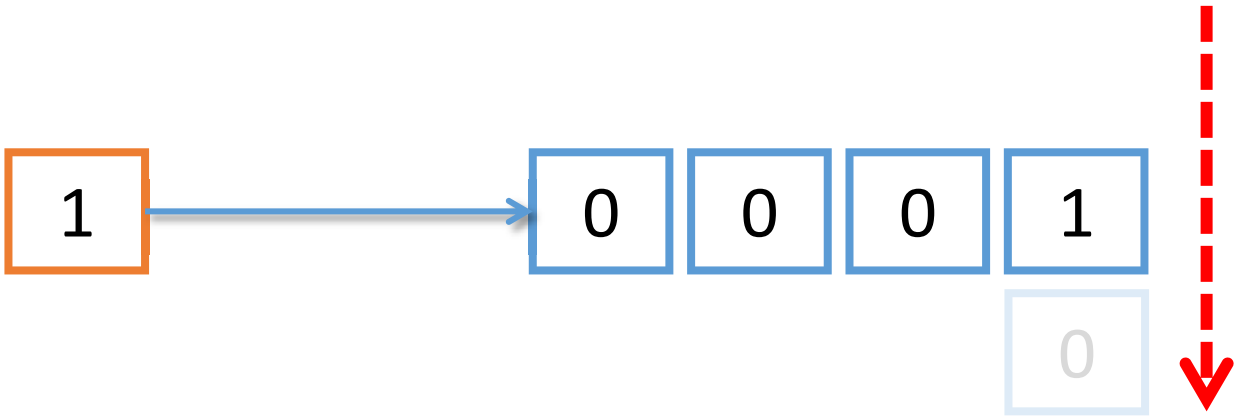
El Sistema de numeración binario

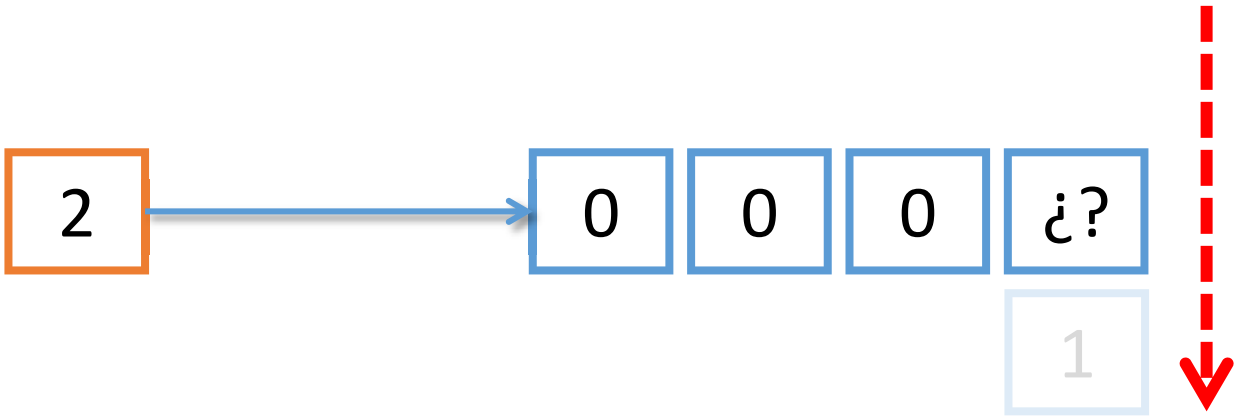
http://pixabay.com/es/gesto-lenguaje-de-se%C3%B1as-dedo-v-422566/

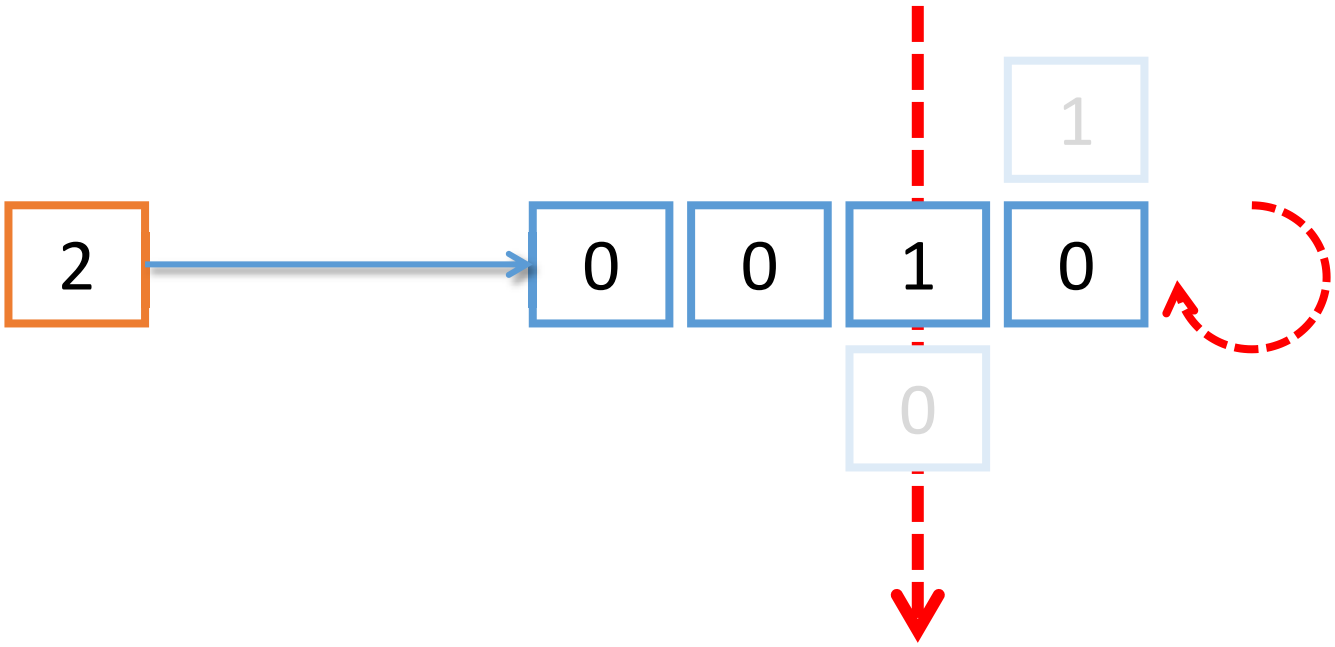
- Es un sistema de numeración **posicional.**
- **Dos** símbolos 0, 1 (b=2).
- Representación de dos estados diferentes de una propiedad física.

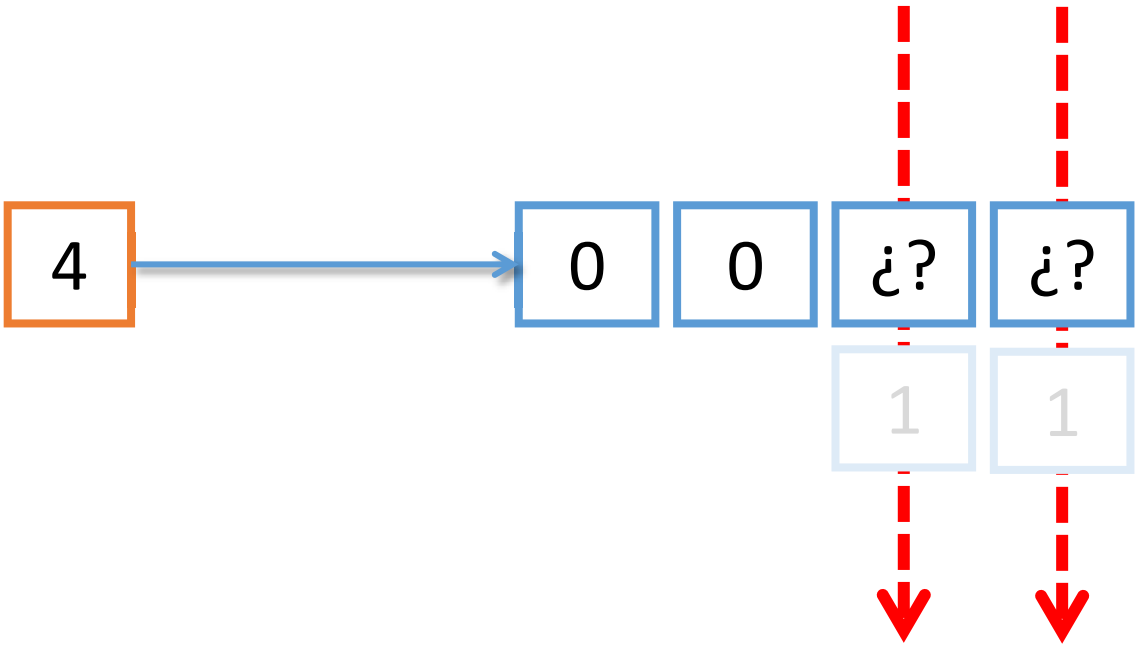
- ¿Cómo funciona el sistema de numeración binario?
- Supongamos N=4
  - N es el número de dígitos que podemos usar

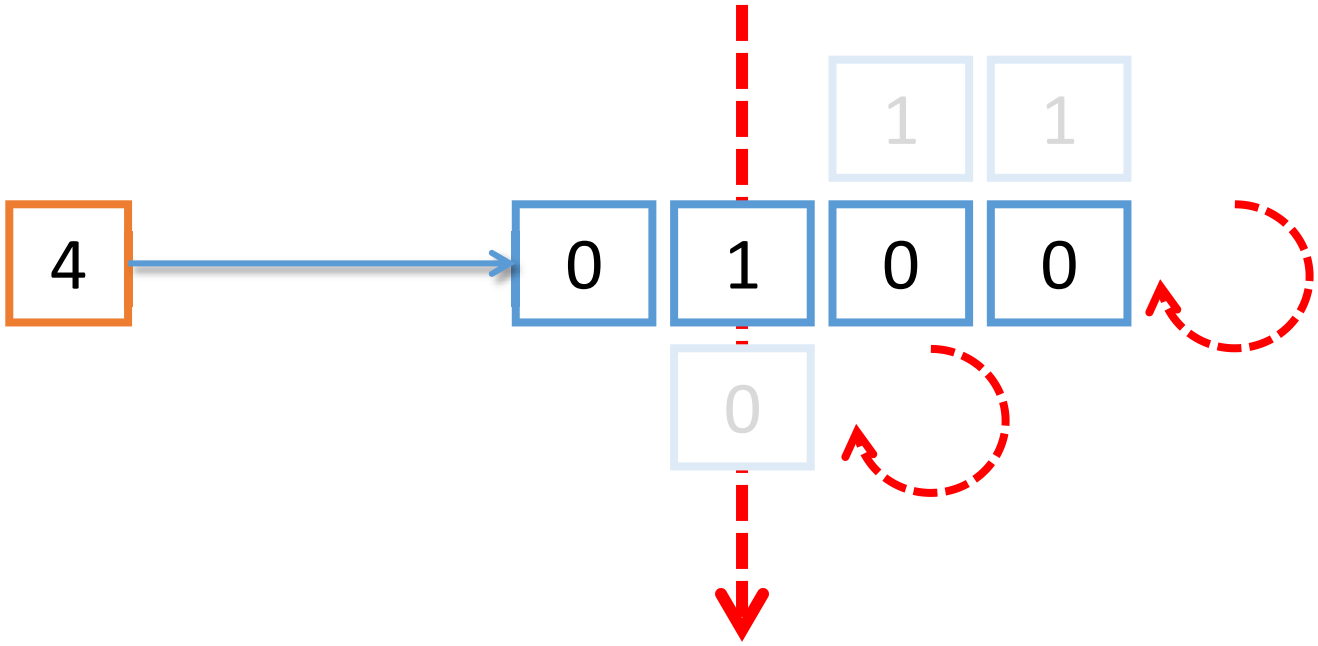


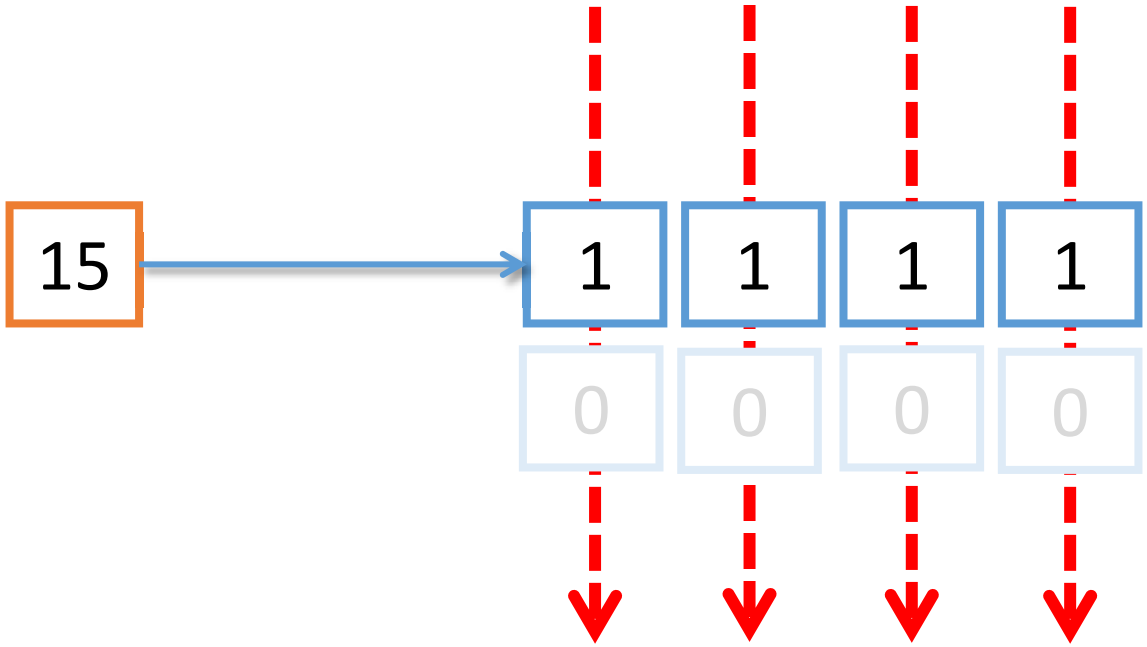

1
0
0
0
1
0


2
0
0
0
¿?
1


2
0
0
1
0
1
0


4
0
0
¿?
¿?
1
1


4
0
1
0
0
1
1
0


15
1
1
1
1
0
0
0
0

- Con N dígitos, es posible representar el rango:

$$[\ 0,\ 2^{N}\ -1]$$

- Con N=4 dígitos:

$$[0,\ 15]$$

El Sistema de numeración binario

De Binario a Decimal

- Aplicar la fórmula:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Con b=2

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

00101_2

- b=2
- N=5
- p=4,3,2,1,0

$x_b[4] = 0$
$x_b[3] = 0$
$x_b[2] = 1$
$x_b[1] = 0$
$x_b[0] = 1$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

00101_2 =

$= x_b[4]*2^4 + x_b[3]*2^3 + x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

00101_2 =

$= x_b[4]*2^4 + x_b[3]*2^3 + x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0$

$= 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

00101_2 =

$= x_b[4]*2^4 + x_b[3]*2^3 + x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0$

$= 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$

$= 0 + 0 + 1*4 + 0 + 1*1$

$= 4 + 1$

$= 5_{10}$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

110_2

- b=2
- N=3
- p=2,1,0

$x_b[2] = 1$
$x_b[1] = 1$
$x_b[0] = 0$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

110_2 =

$= x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

110_2 =

$= x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0$

$= 1*2^2 + 1*2^1 + 0*2^0$

Ejemplo:

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

110_2 =

$= x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0$

$= 1*2^2 + 1*2^1 + 0*2^0$

$= 4 \quad + 2 \quad + 0$

$= 4 + 2$

$= 6_{10}$

El Sistema de numeración binario

De Decimal a Binario

- Dos métodos:
  - El método de las divisiones sucesivas por la base
  - El método de las potencias por la base

Método de las divisiones por la base

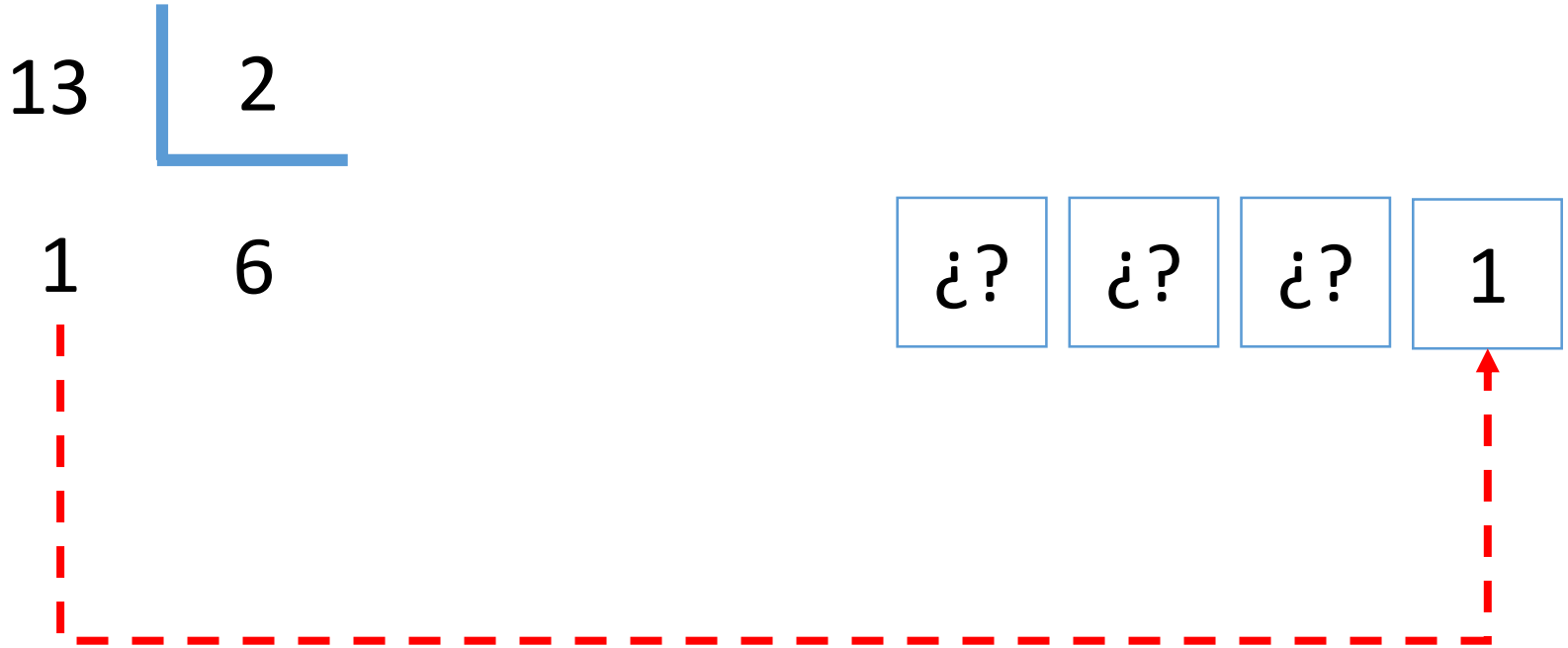
Método divisiones por la base

- Convertir 13_{10} a binario

13	2
1	6

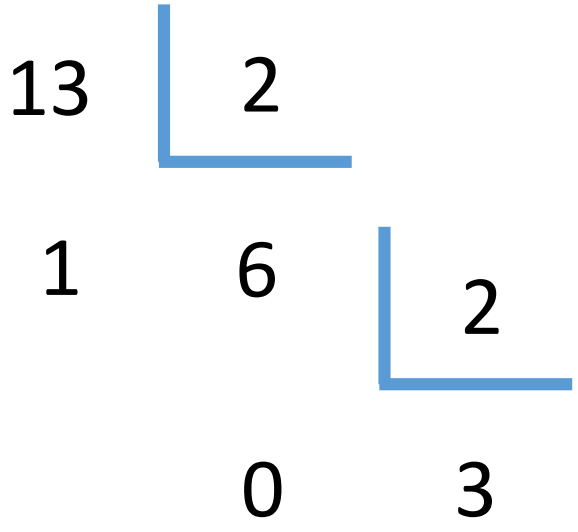
Método divisiones por la base

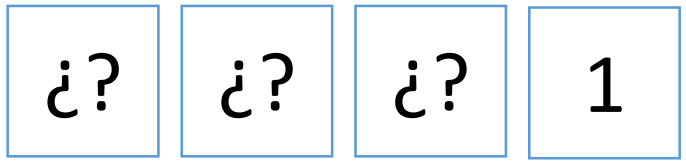
- Convertir 13_{10} a binario



Método divisiones por la base

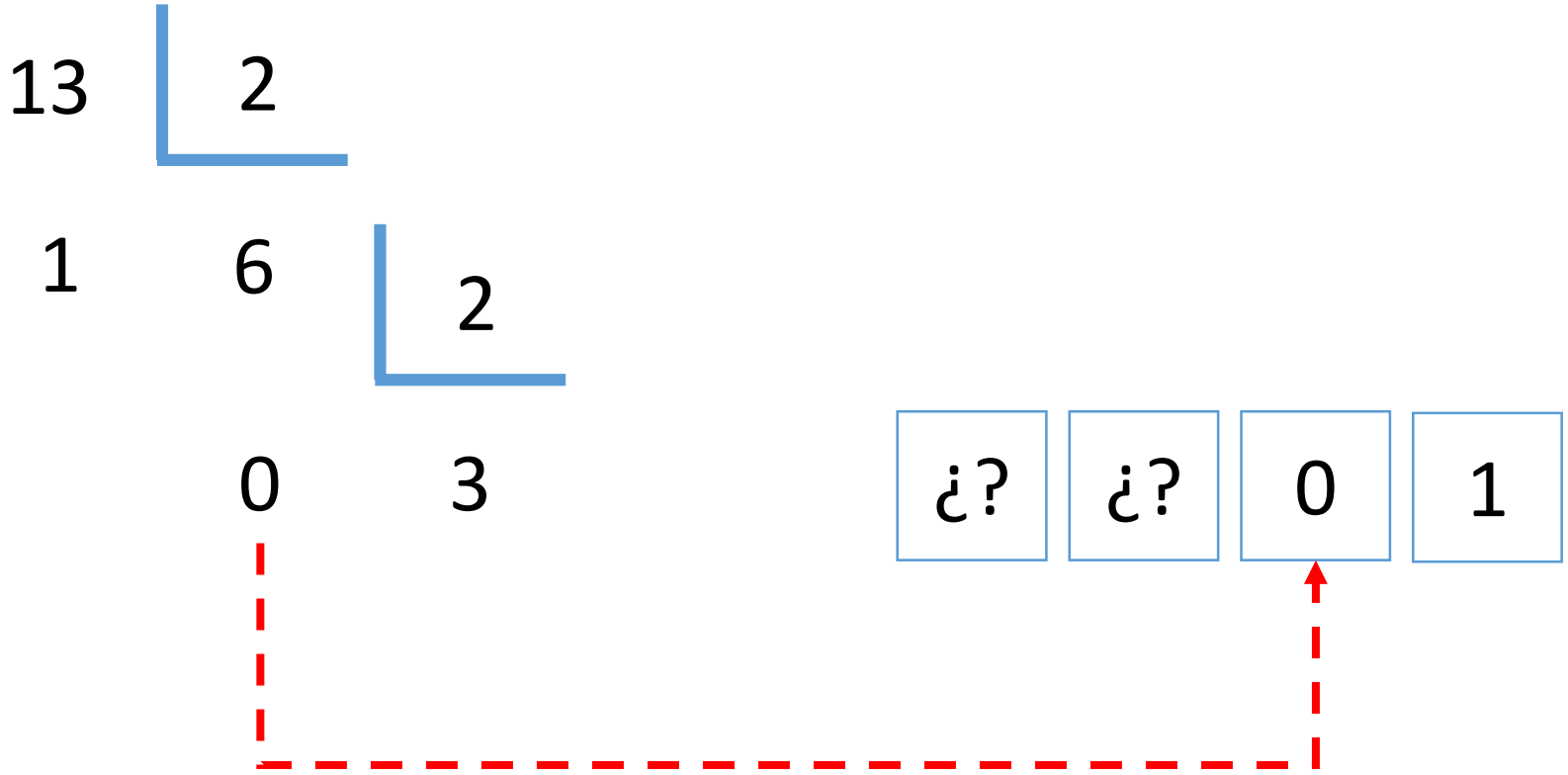
- Convertir 13_{10} a binario





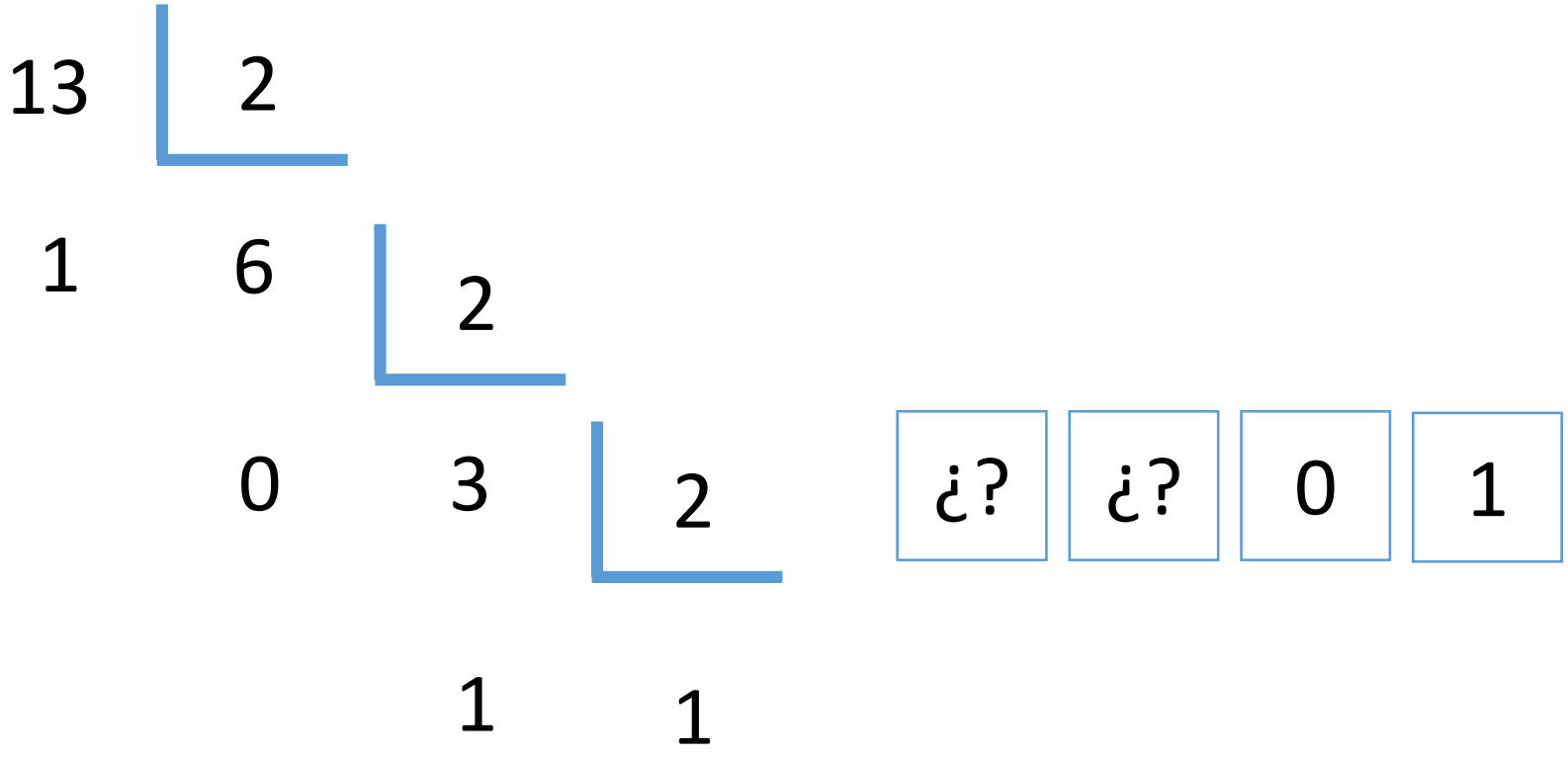
Método divisiones por la base

- Convertir 13_{10} a binario



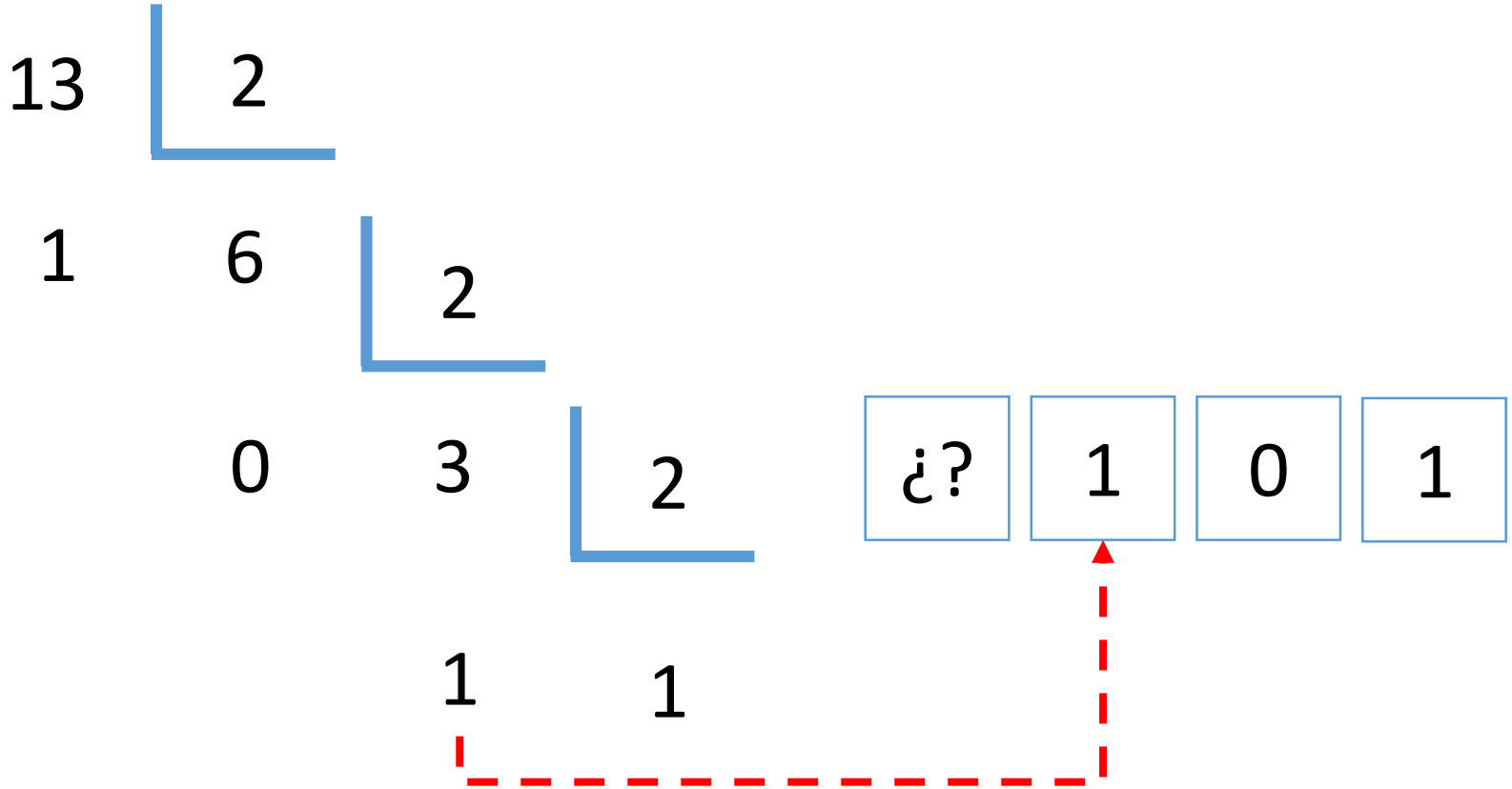
Método divisiones por la base

- Convertir 13_{10} a binario



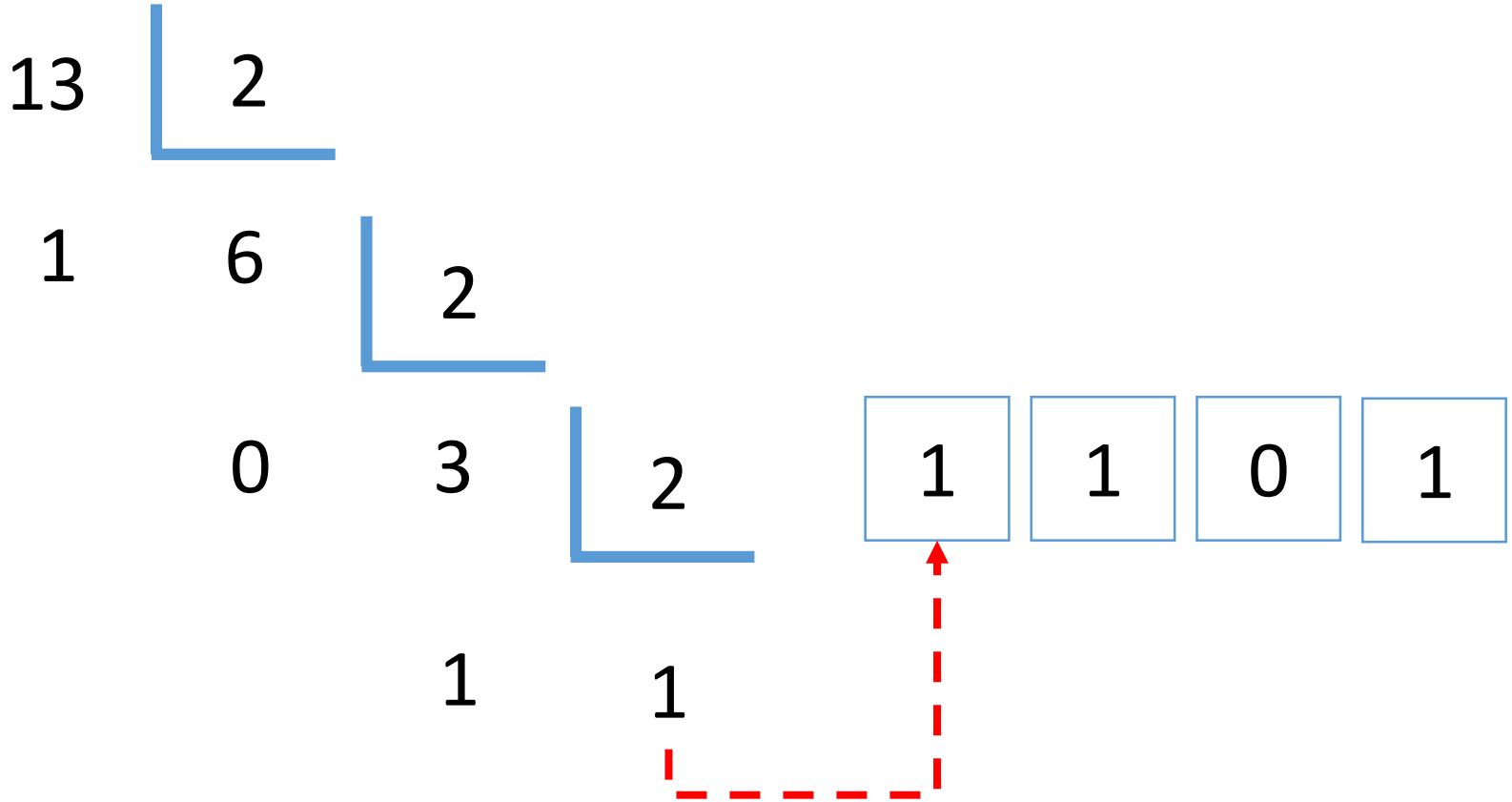
Método divisiones por la base

- Convertir 13_{10} a binario



Método divisiones por la base

- Convertir 13_{10} a binario



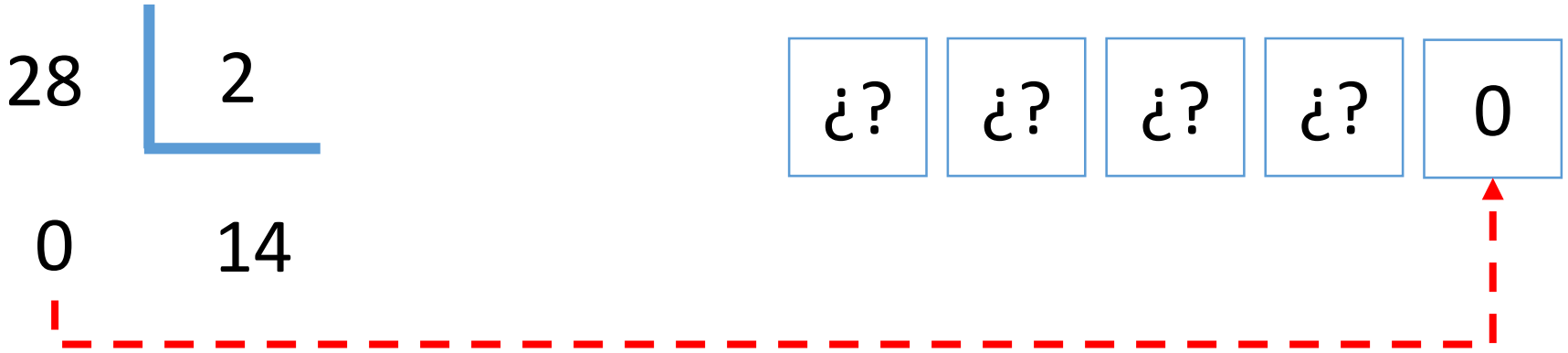
Método divisiones por la base

- Convertir 28_{10} a binario

28	2
0	14

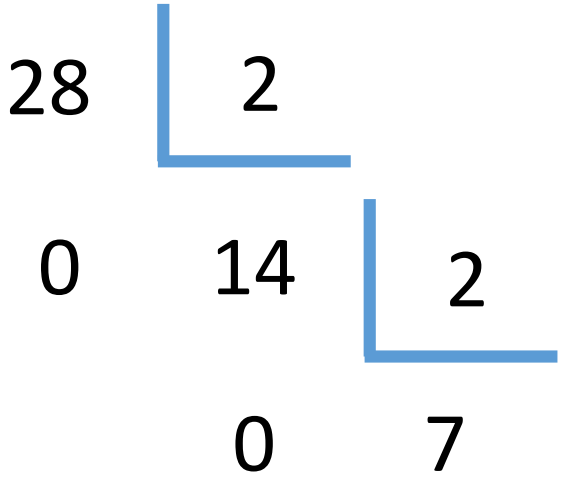
Método divisiones por la base

- Convertir 28_{10} a binario



Método divisiones por la base

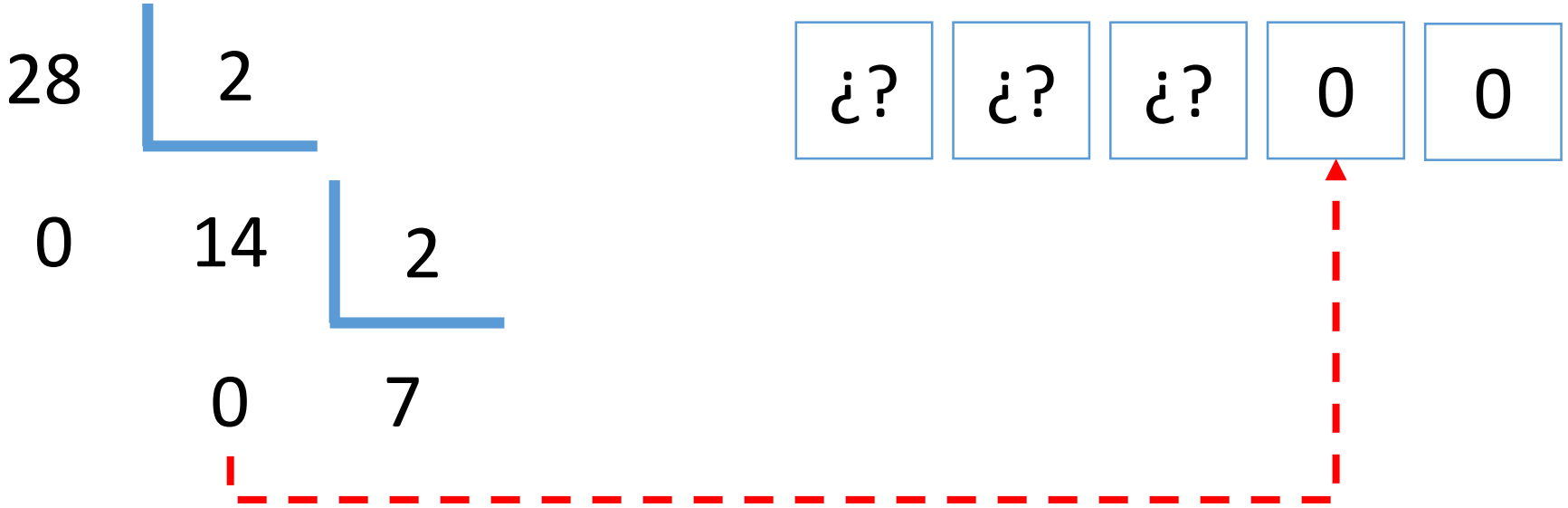
- Convertir 28_{10} a binario



¿?	¿?	¿?	¿?	0

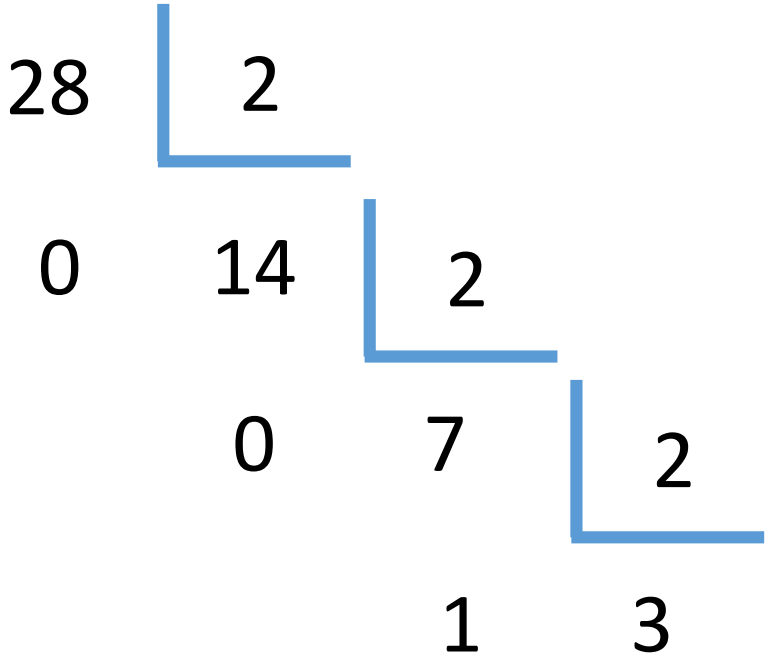
Método divisiones por la base

- Convertir 28_{10} a binario



Método divisiones por la base

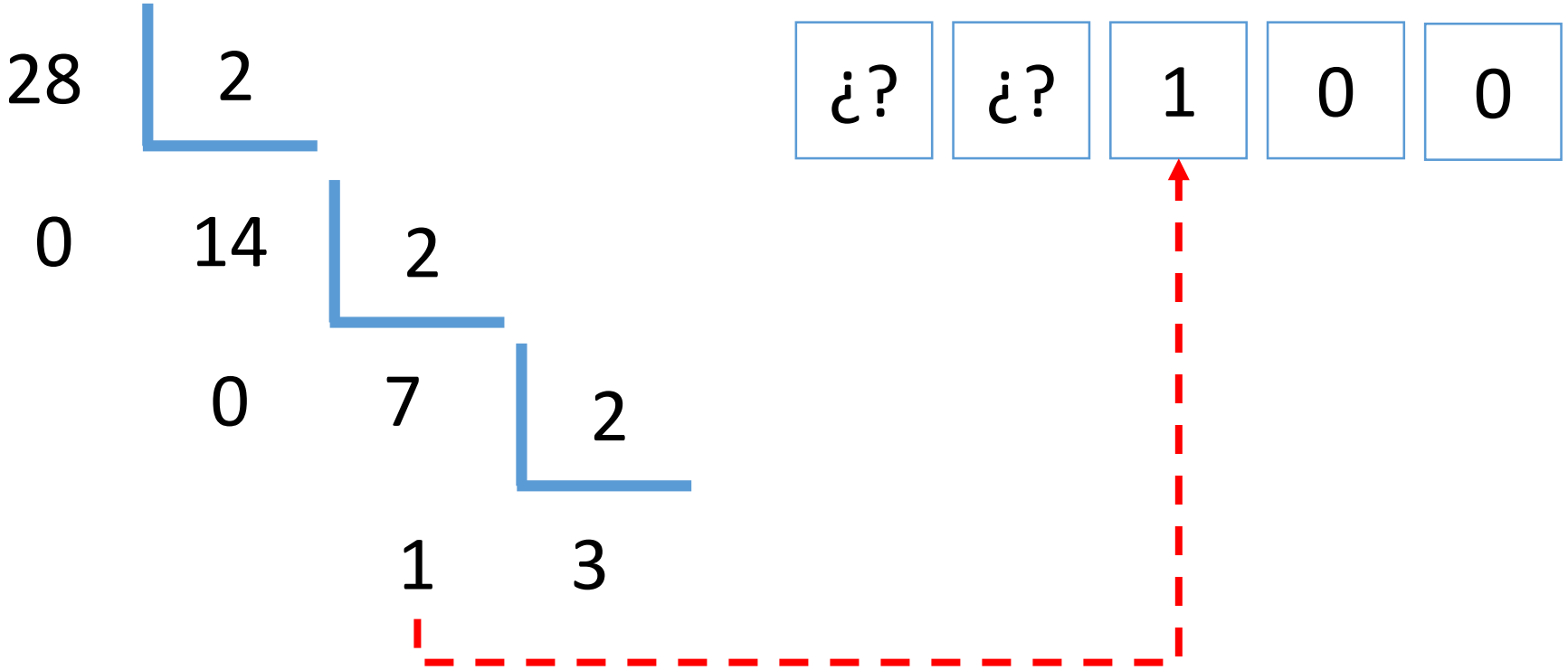
- Convertir 28_{10} a binario



¿?	¿?	¿?	0	0

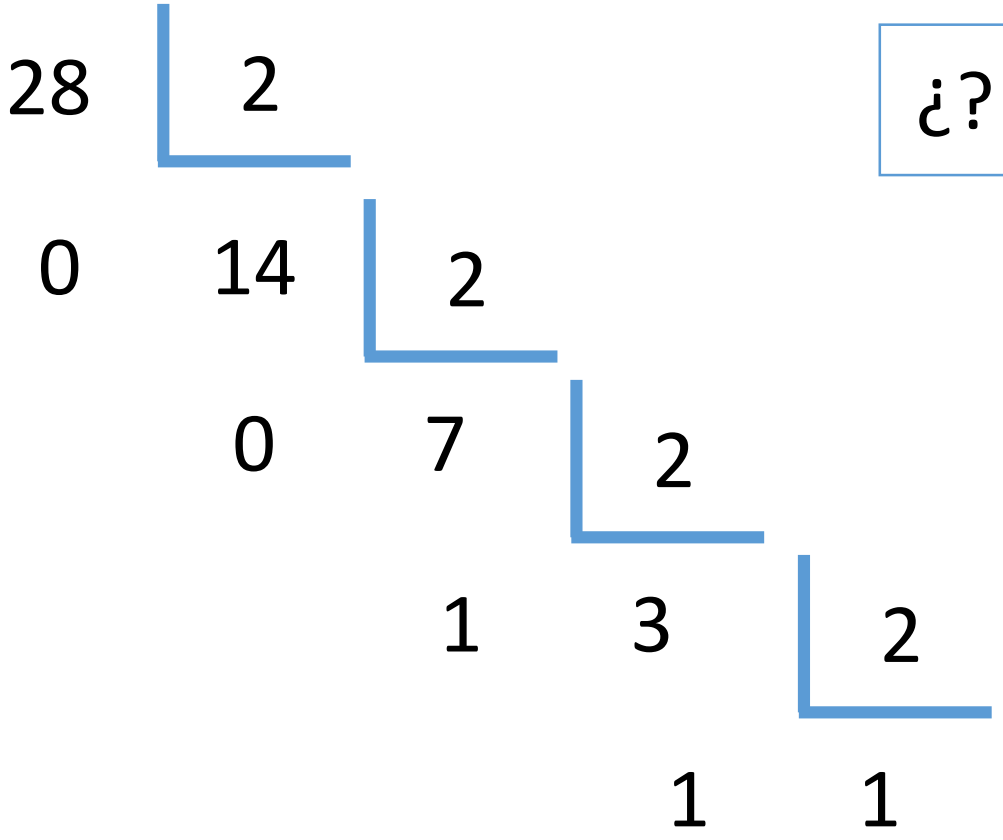
Método divisiones por la base

- Convertir 28_{10} a binario



Método divisiones por la base

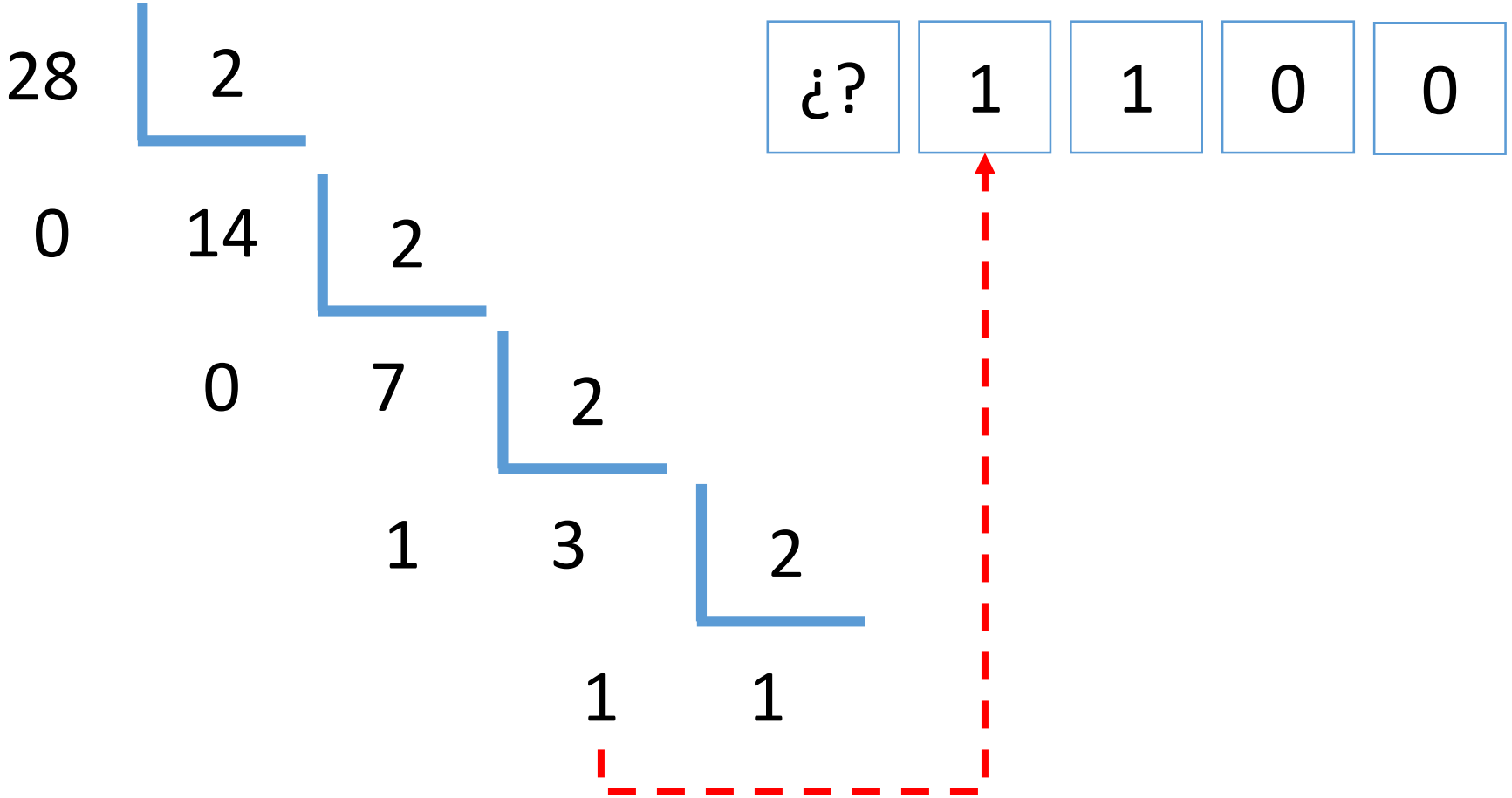
- Convertir 28_{10} a binario



¿?	¿?	1	0	0

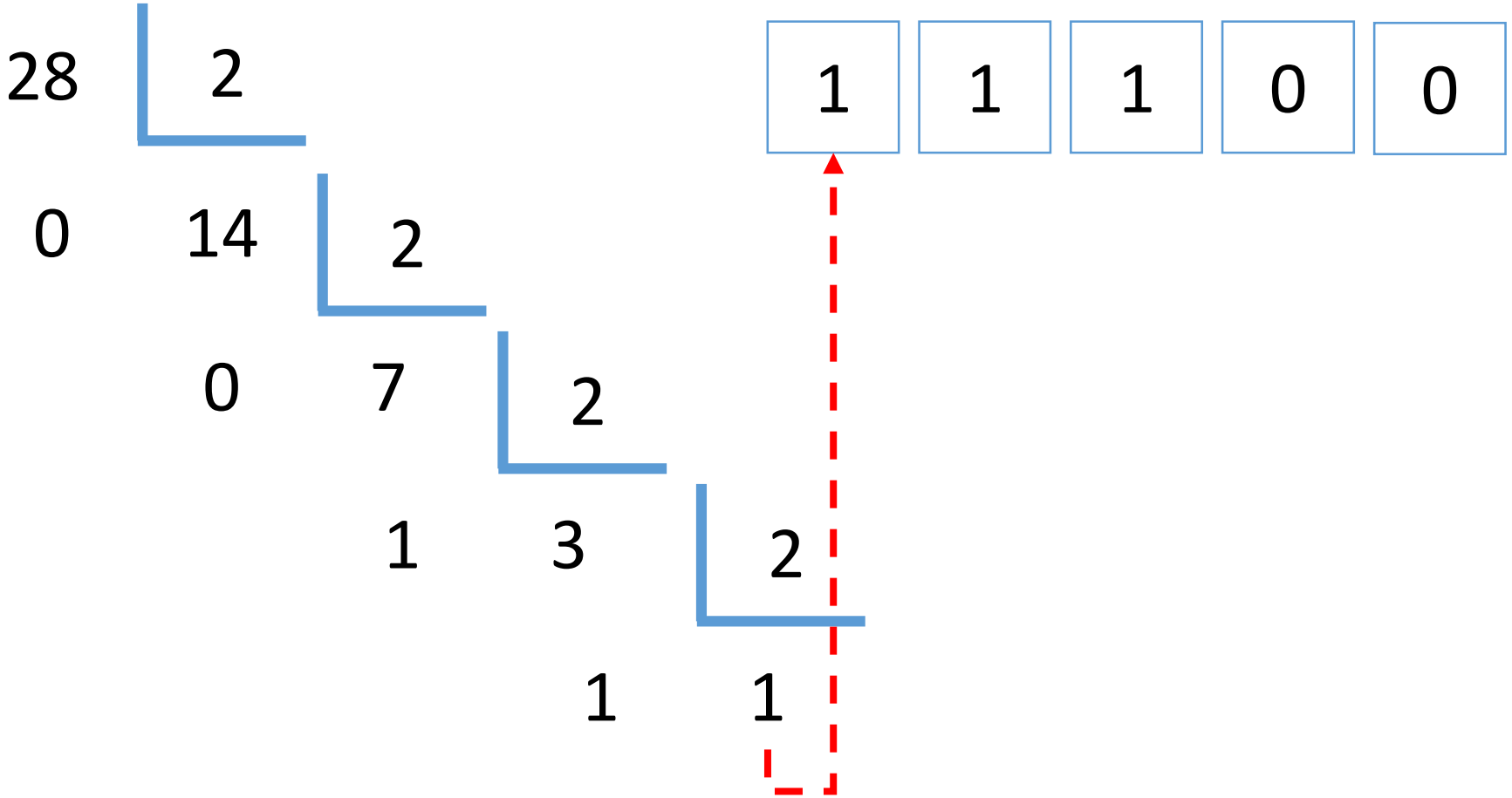
Método divisiones por la base

- Convertir 28_{10} a binario



Método divisiones por la base

- Convertir 28_{10} a binario



Método de las potencias de la base

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
2^9	512

- Convertir 43_{10} a binario

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
2^9	512

- Convertir 43_{10} a binario

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
2^9	512

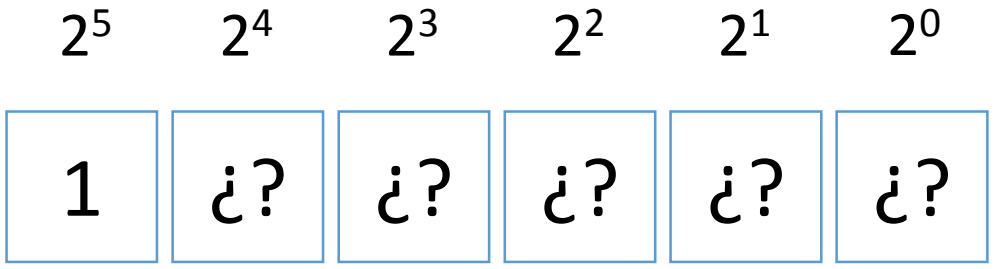
$$x_b[5]*2^5 + x_b[4]*2^4 + x_b[3]*2^3 + x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0$$

$$x_b[5] = 1$$

2^5	2^4	2^3	2^2	2^1	2^0
1	¿?	¿?	¿?	¿?	¿?

- Convertir 43_{10} a binario
  - 43 – 32 = 11

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
2^9	512



- Convertir 43_{10} a binario
  - 43 – 32 = 11

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
2^9	512

ATENCIÓN: No es 2^4

$x_b[4] = 0$

2^5	2^4	2^3	2^2	2^1	2^0
1	0	¿?	¿?	¿?	¿?

- Convertir 43_{10} a binario
  - 43 – 32 = 11

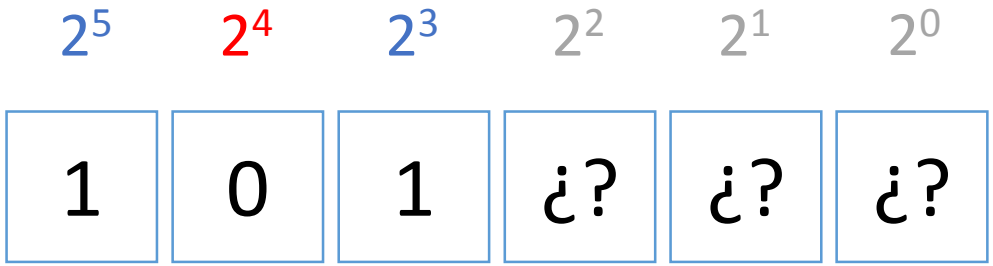
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512

$x_b[3] = 1$

2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	¿?	¿?	¿?

- Convertir 43_{10} a binario
  - 43 – 32 = 11
  - 11 – 8 = 3

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
2^9	512



- Convertir 43_{10} a binario
  - 43 – 32 = 11
  - 11 – 8 = 3

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
2^9	512

ATENCIÓN: No es 2^2

$x_b[2] = 0$

2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	¿?	¿?

- Convertir 43_{10} a binario
  - 43 – 32 = 11
  - 11 – 8 = 3

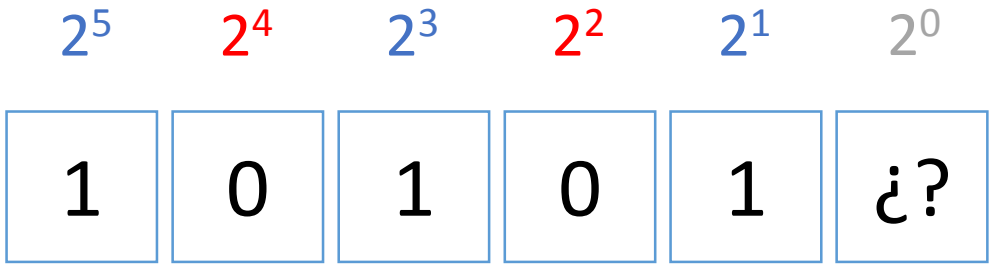
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512

$x_b[1] = 1$

2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	¿?

- Convertir 43_{10} a binario
  - 43 – 32 = 11
  - 11 – 8 = 3
  - 3 – 2 = 1

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
2^9	512



- Convertir 43_{10} a binario
  - 43 – 32 = 11
  - 11 – 8 = 3
  - 3 – 2 = 1

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
2^9	512

$x_b[0] = 1$

2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	1

- Convertir 28_{10} a binario

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
2^9	512

- Convertir 28_{10} a binario

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
2^9	512

$$x_b[4]*2^4 + x_b[3]*2^3 + x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0$$

$$x_b[4] = 1$$

2^4	2^3	2^2	2^1	2^0
1	¿?	¿?	¿?	¿?

- Convertir 28_{10} a binario
  - 28 – 16 = 12

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
2^9	512

$x_b[3] = 1$

2^4	2^3	2^2	2^1	2^0
1	1	¿?	¿?	¿?

- Convertir 28_{10} a binario
  - 28 – 16 = 8
  - 12 – 8 = 4

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
2^9	512

$x_b[2] = 1$

2^4	2^3	2^2	2^1	2^0
1	1	1	¿?	¿?

- Convertir 28_{10} a binario
  - 28 – 16 = 8
  - 12 – 8 = 4
  - 4 – 4 = 0

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
2^9	512

Como es 0, el resto 0

$x_b[1] = 0$

$x_b[0] = 0$

2^4	2^3	2^2	2^1	2^0
1	1	1	0	0

El Sistema de numeración binario

Números reales

- Número real X = Y + Z:
  - Y: parte entera
    - Lo que hay a la izquierda de la coma.
  - Z: parte fraccionaria:
    - Lo que hay a la derecha de la coma.

- Número real X = Y + Z:
  - Y: parte entera
    - Lo que hay a la izquierda de la coma.
  - Z: parte fraccionaria:
    - Lo que hay a la derecha de la coma.
  - X = 5,65
    - Parte entera: Y = 5
    - Parte fraccionaria: Z= 0,65

- ATENCIÓN:
  - Los ordenadores **NO** representan los números reales como por ejemplo $110{,}100_2$
    - NO existe la coma en binario

- ATENCIÓN:
  - Los ordenadores **NO** representan los números reales como por ejemplo $110{,}100_2$
    - NO existe la coma en binario
  - Se usa el formato **IEEE754**
    - Es necesario saber como convertir la parte fraccionaria de un número real en decimal a binario y viceversa.

De Binario a Decimal

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

De binario a decimal:

$101{,}101_2$

- b=2
- N+=3
- N-=3
- p=2,1,0
- q=-1,-2,-3

$x_b[2] = 1$
$x_b[1] = 0$
$x_b[0] = 1$

$x_b[-1] = 1$
$x_b[-2] = 0$
$x_b[-3] = 1$

$$x_{10} = \sum_{p=N^{+}-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^{-}} x_b[q] * b^q$$

De binario a decimal:

$101{,}101_2 =$

$= \quad x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0 +$

$x_b[-1]*2^{-1} + x_b[-2]*2^{-2} + x_b[-3]*2^{-3}$

$$x_{10} = \sum_{p=N^+-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^-} x_b[q] * b^q$$

De binario a decimal:

$101{,}101_2 =$

$= \quad x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0 +$

$x_b[-1]*2^{-1} + x_b[-2]*2^{-2} + x_b[-3]*2^{-3}$

$= \quad 1*2^2 + 0*2^1 + 1*2^0 +$

$1*2^{-1} + 0*2^{-2} + 1*2^{-3}$

$$x_{10} = \sum_{p=N^{+}-1}^{0} x_b[p] * b^p + \sum_{q=-1}^{-N^{-}} x_b[q] * b^q$$

De binario a decimal:

$101{,}101_2 =$

$= x_b[2]*2^2 + x_b[1]*2^1 + x_b[0]*2^0 +$

$x_b[-1]*2^{-1} + x_b[-2]*2^{-2} + x_b[-3]*2^{-3}$

$= 1*2^2 + 0*2^1 + 1*2^0 +$

$1*2^{-1} + 0*2^{-2} + 1*2^{-3}$

$= 4 + 0 + 1 + 0{,}5 + 0{,}125$

$= 5{,}625_{10}$

De Decimal a Binario

- ATENCIÓN:
  - Parte entera:
    - Método **divisiones** sucesivas por la base
    - Método potencias **positivas** de la base
  - Parte fraccionaria:
    - Método **multiplicaciones** sucesivas por la base
    - Método potencias **negativas** de la base

- ATENCIÓN:
  - Parte entera:
    - Método **divisiones** sucesivas por la base
    - Método potencias **positivas** de la base
  - Parte fraccionaria:
    - Método **multiplicaciones** sucesivas por la base
    - Método potencias **negativas** de la base

- Convertir $5{,}25_{10}$:
  - Parte entera:
    - $5_{10} \equiv 101_2$
  - Parte fraccionaria:
    - Aplicamos el método de las **multiplicaciones** sucesivas por la base a $0{,}25_{10}$

- Convertir $0,25_{10}$:

0,25

X 2

0 , ¿? ¿?

- Convertir $0,25_{10}$:

- Convertir $0,25_{10}$:

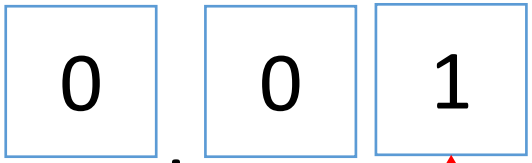
0,25

X 2

0,50

X 2

1,00



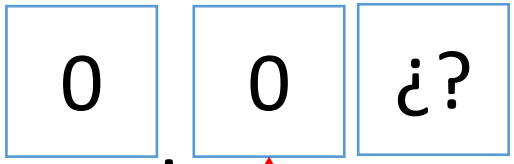
- Convertir $7{,}4_{10}$:
  - Parte entera:
    - $7_{10} \equiv 111_2$
  - Parte fraccionaria:
    - Aplicamos el método de las **multiplicaciones** sucesivas por la base a $0{,}4_{10}$

0	,	¿?	¿?

- Convertir $0,4_{10}$:

0,40

X 2



- Convertir $0,4_{10}$:

$$\begin{array}{r} 0,40 \\ X \quad 2 \\ \hline 0,80 \end{array}$$

- Convertir $0,4_{10}$:

0,40

X 2

0,80

X 2

1,60

- Pero $0{,}4_{10}$ no es igual a $0{,}01_2$:

- $0{,}01_2 = 0{,}25_{10}$

- Pero $0{,}4_{10}$ no es igual a $0{,}01_2$:

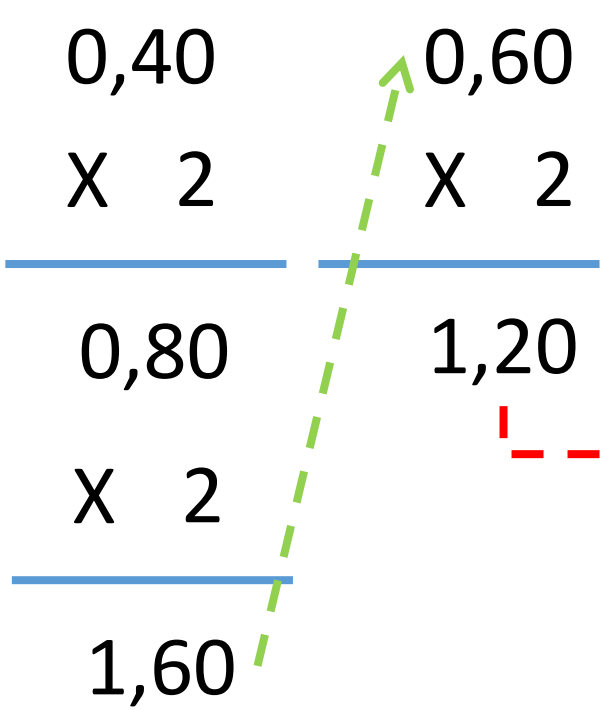
- $0{,}01_2 = 0{,}25_{10}$

- Hemos obtenido solo una aproximación.

- Podemos mejorar la aproximación siguiendo el proceso.

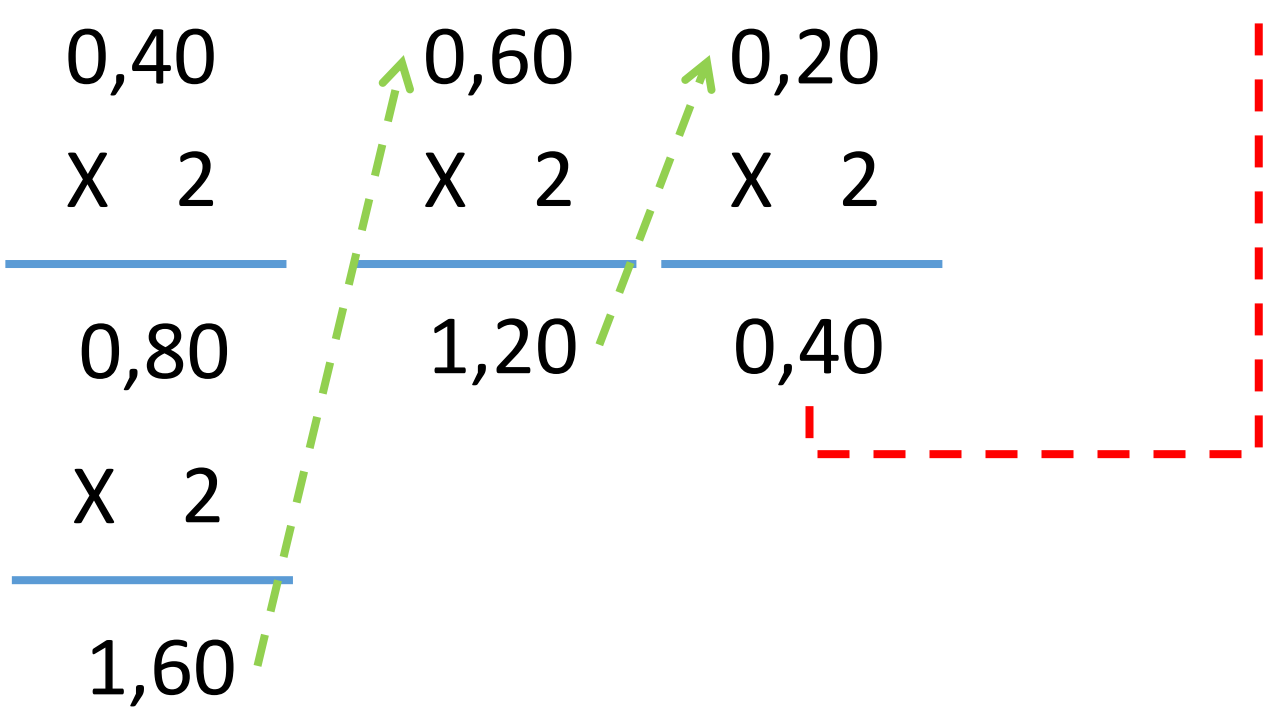
0	,	0	1	1	¿?	¿?	¿?

- Convertir $0,4_{10}$:



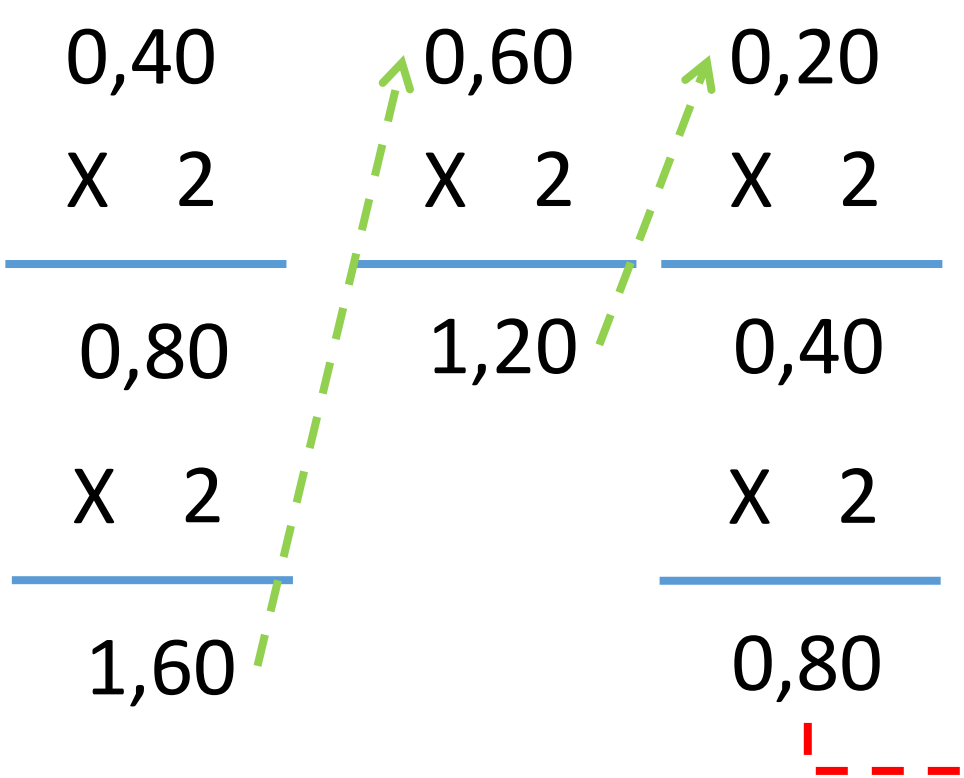
0	,	0	1	1	0	¿?	¿?

- Convertir $0,4_{10}$:



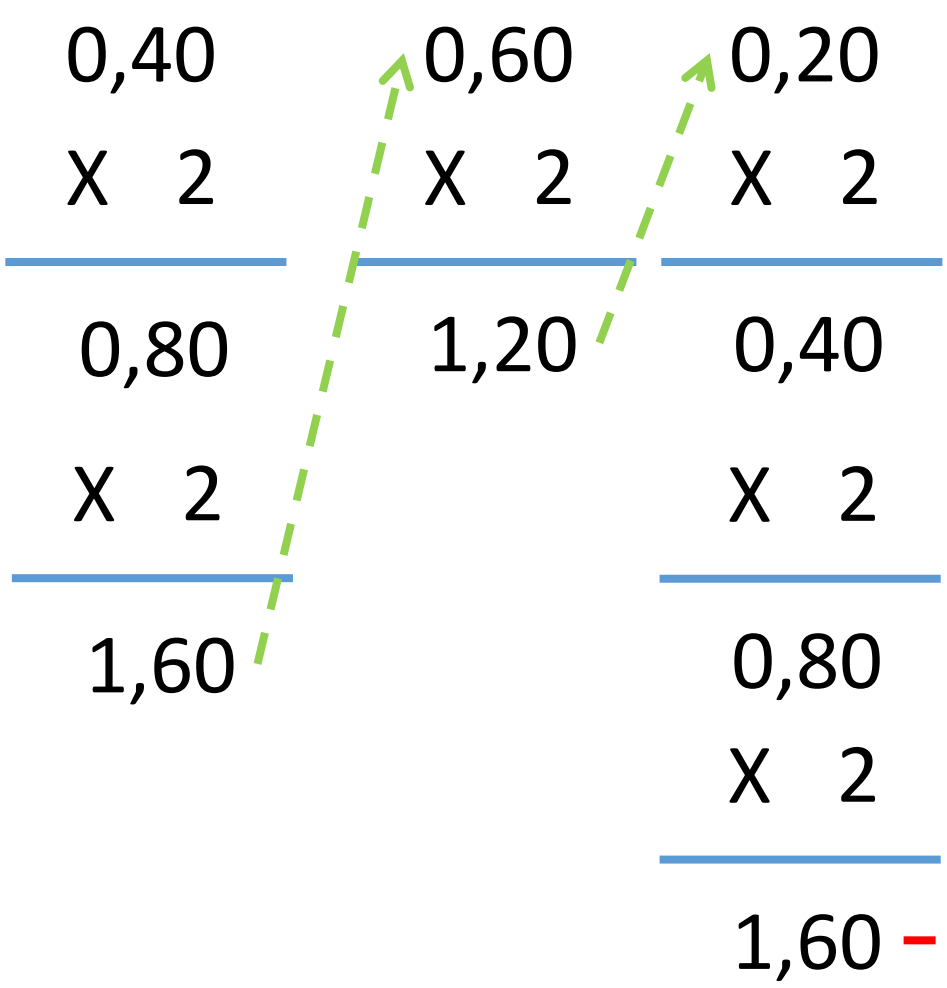
0	0	1	1	0	0	¿?

- Convertir $0,4_{10}$:



0	,	0	1	1	0	0	1

- Convertir $0,4_{10}$:



- $0{,}4_{10}$ es aproximadamente igual a $0{,}011001_2$
- Exactamente $0{,}011001_2 \equiv 0{,}3906_{10}$
- Cuantos más dígitos, mejor aproximación.

El Sistema de numeración binario

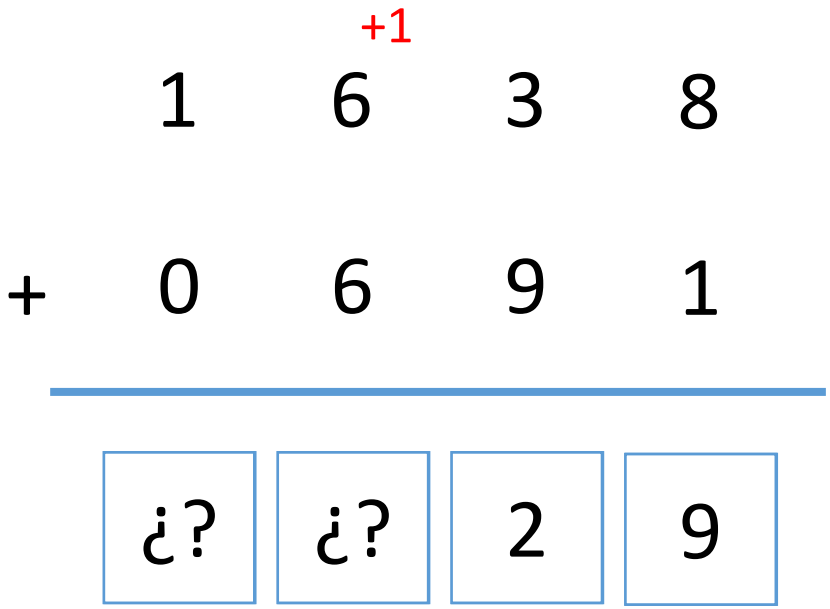
Operar en binario

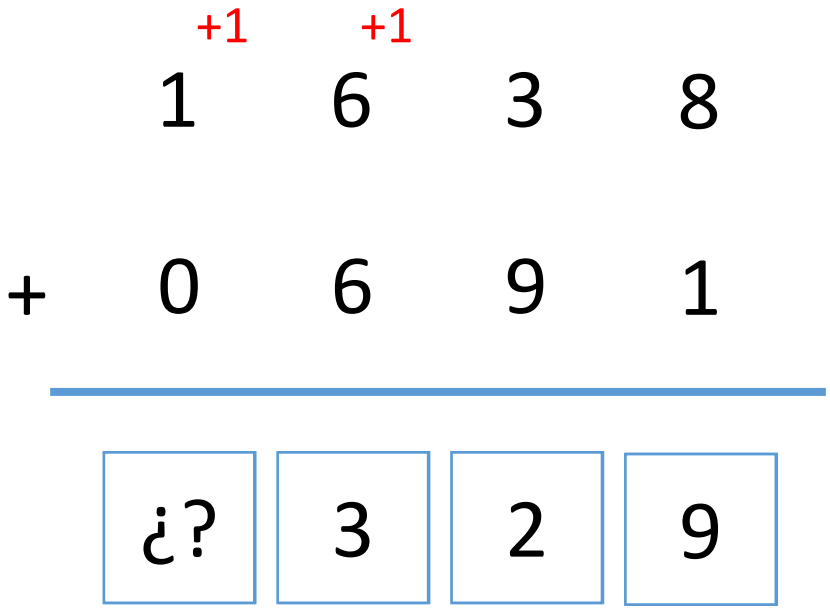
Sumar

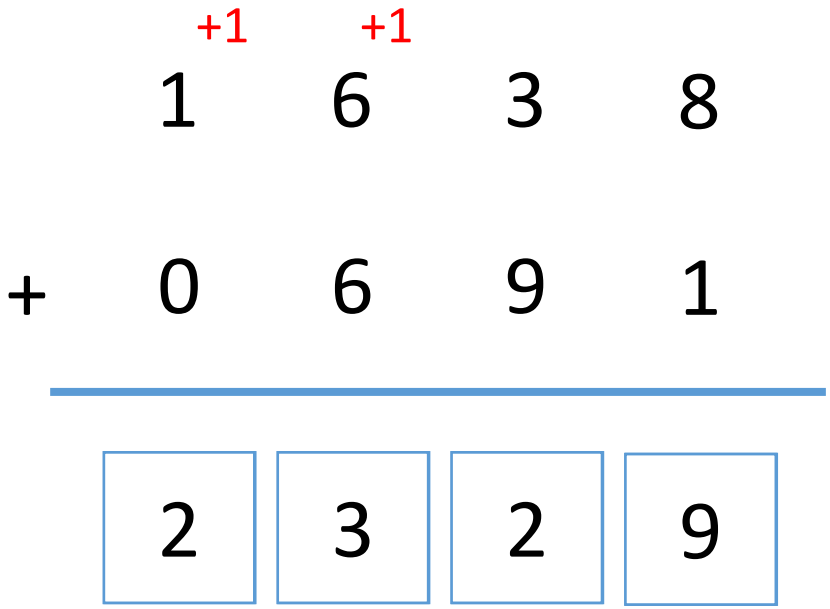
- Hay que saber:
  - El símbolo resultado de la suma de otros dos
  - Si hay acarreo (“llevo 1”) o no

	1	6	3	8
+	0	6	9	1
	¿?	¿?	¿?	¿?

	1	6	3	8
+	0	6	9	1
	¿?	¿?	¿?	9

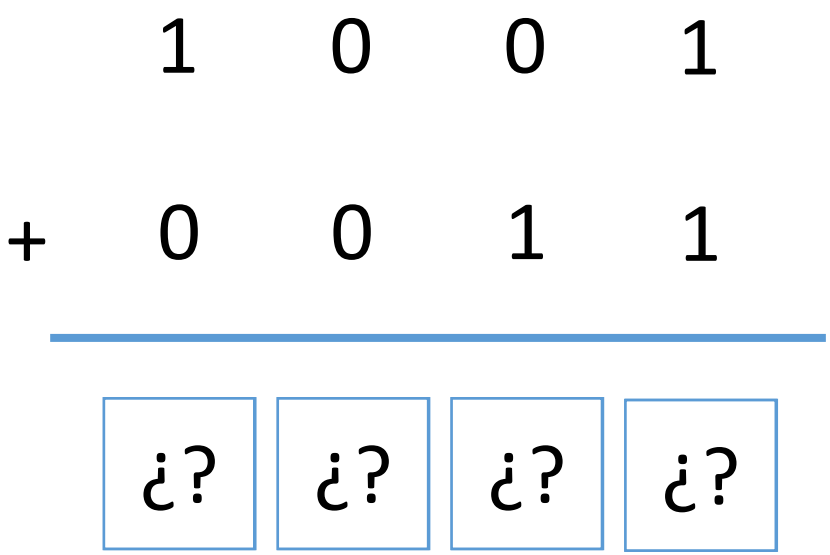

+1
1 6 3 8
+ 0 6 9 1
¿? ¿? 2 9

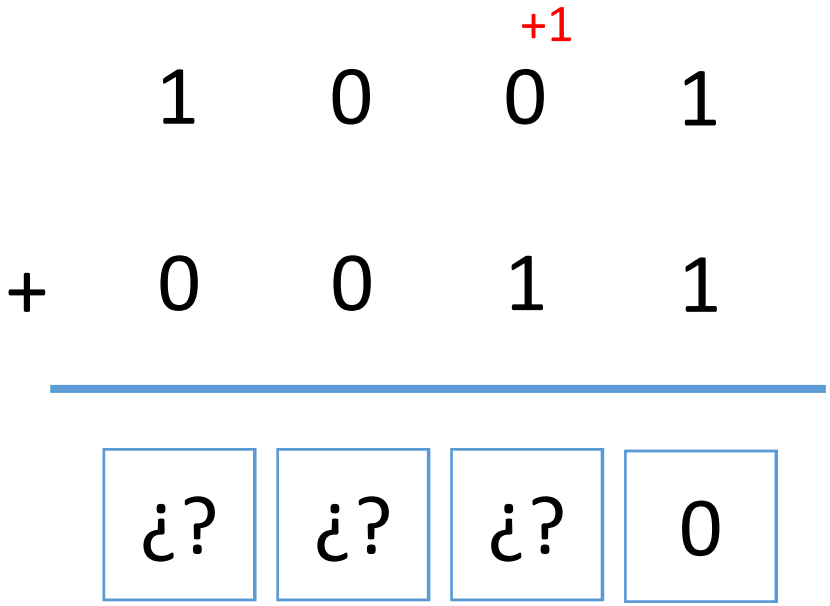

+1
+1
1 6 3 8
+ 0 6 9 1
¿? 3 2 9

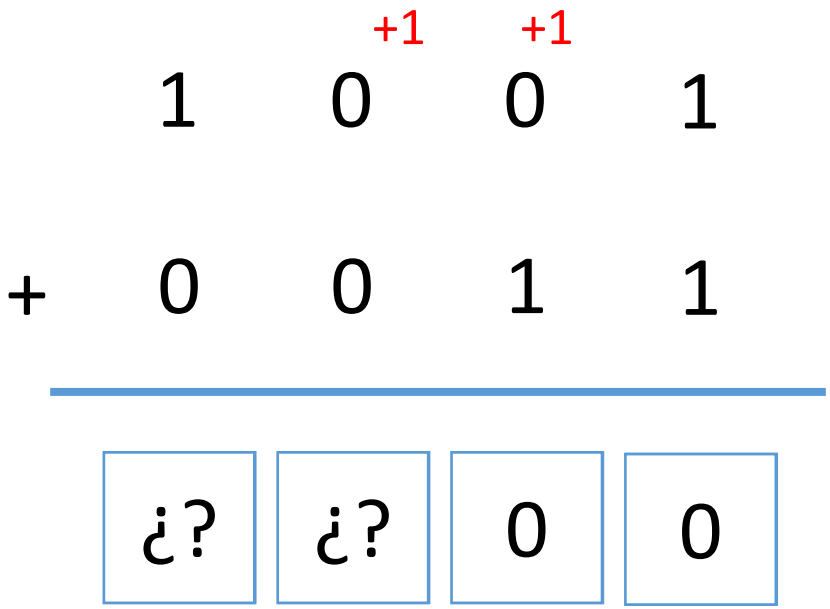

+1
+1
1 6 3 8
+ 0 6 9 1
2 3 2 9

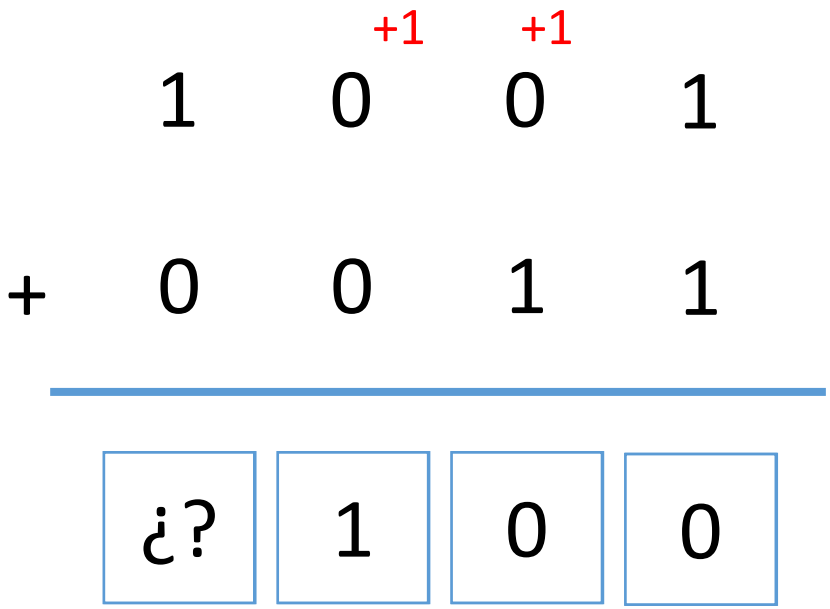
- En binario:

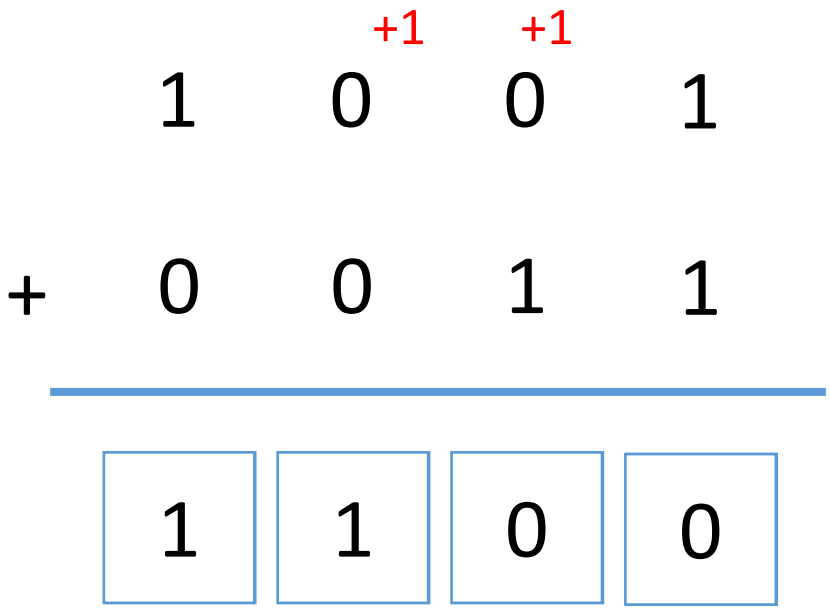
Primer símbolo	Segundo símbolo	Símbolo resultado	Acarreo
0	0	0	No
0	1	1	No
1	0	1	No
1	1	0	Si


1 0 0 1
+ 0 0 1 1
¿? ¿? ¿? ¿?

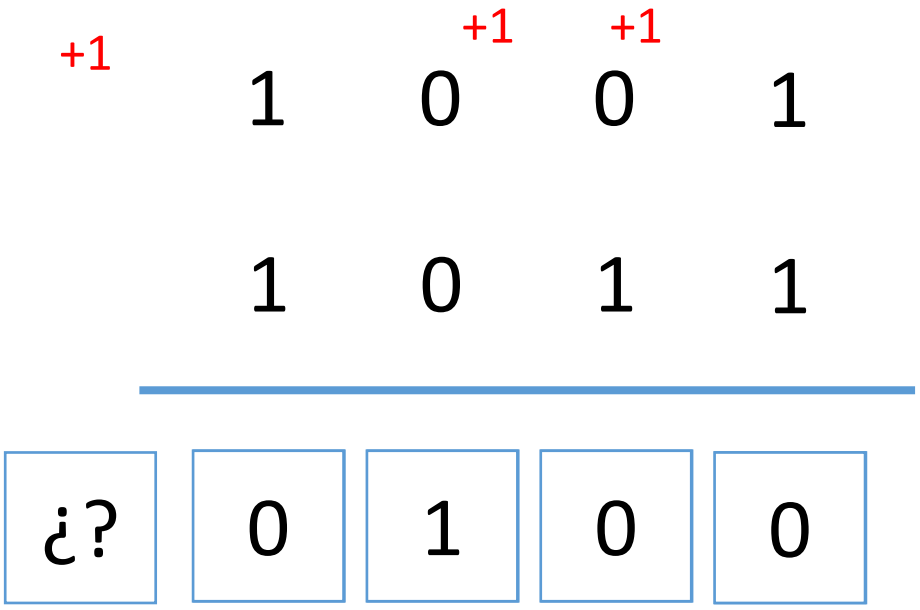

+1
1 0 0 1
+ 0 0 1 1
¿? ¿? ¿? 0


+1
+1
1 0 0 1
+ 0 0 1 1
¿? ¿? 0 0


+1
+1
1 0 0 1
+ 0 0 1 1
¿?
1
0
0


+1
+1
1 0 0 1
+ 0 0 1 1
1 1 0 0

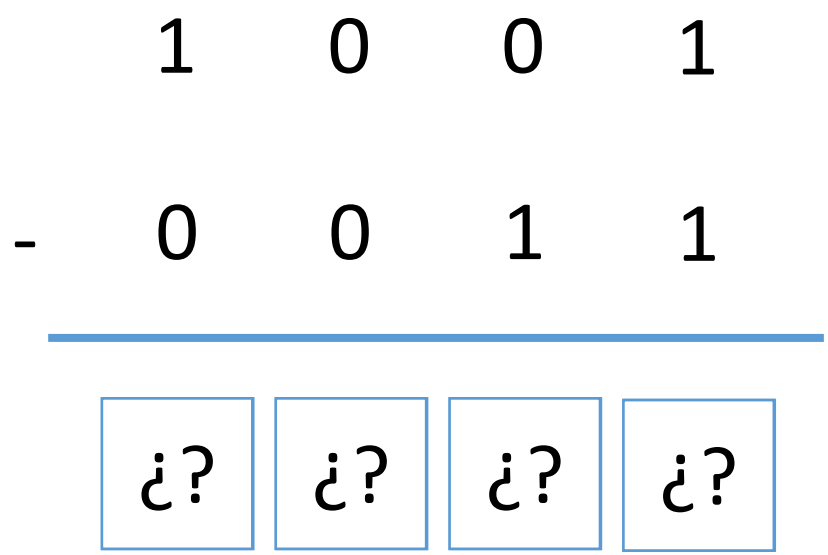
- ¿Qué ocurre si la suma de los dos dígitos más representativos produce acarreo?
  - Desbordamiento

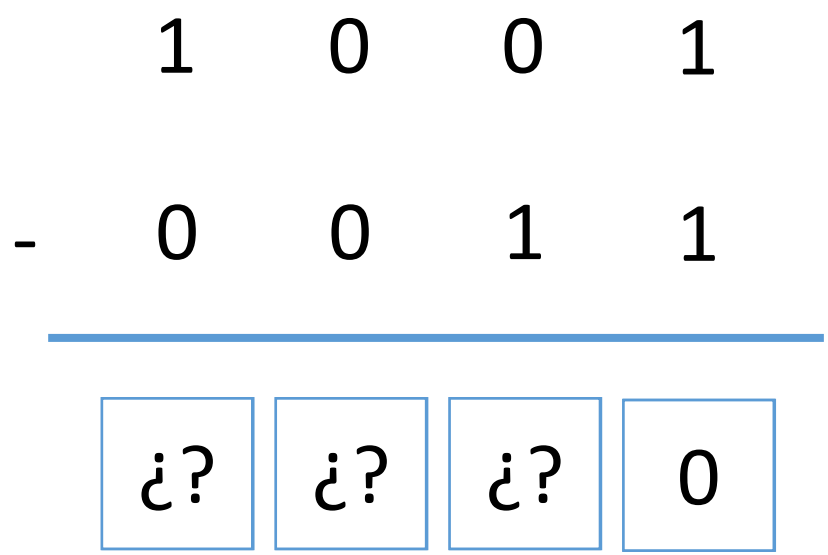


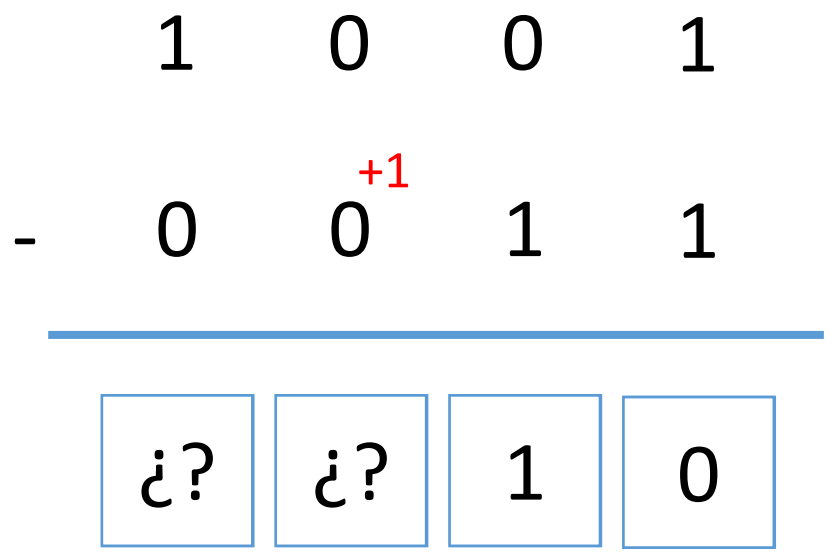
Restar

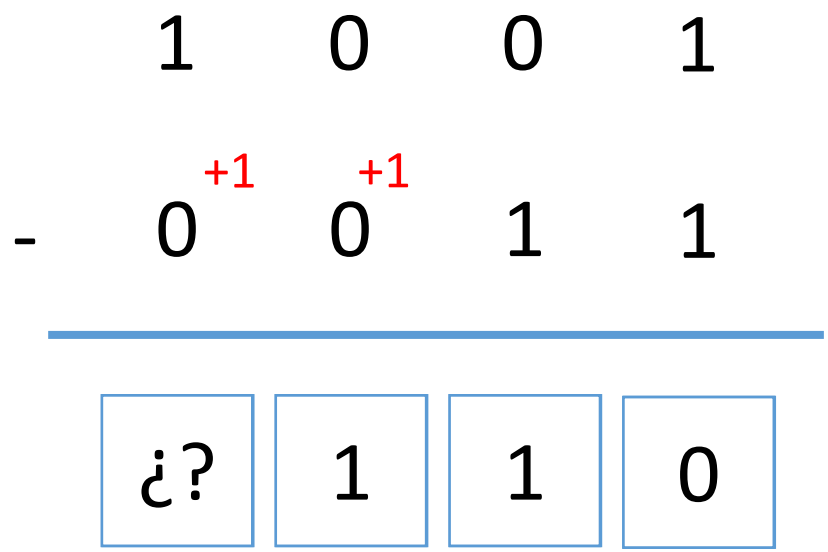
- Hay que saber:
  - El símbolo resultado de la resta de otros dos
  - Si hay acarreo o no
    - Acarreo +1 se suma al de abajo (substraendo)
    - O acarreo -1 que se suma al de arriba (minuendo)

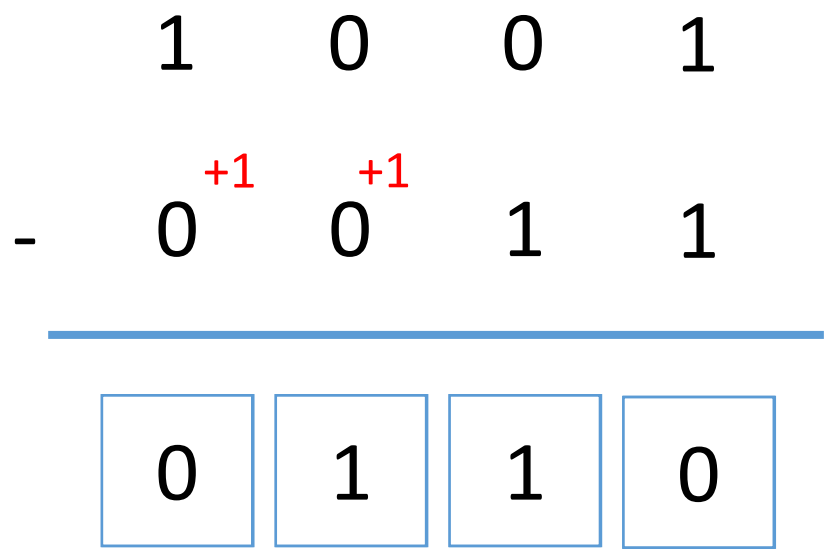
Primer símbolo	Segundo símbolo	Símbolo resultado	Acarreo
0	0	0	No
0	1	1	Si
1	0	1	No
1	1	0	No


1 0 0 1
- 0 0 1 1
¿? ¿? ¿? ¿?


1 0 0 1
- 0 0 1 1
¿? ¿? ¿? 0


1 0 0 1
+1
- 0 0 1 1
¿? ¿? 1 0


1 0 0 1
+1 +1
- 0 0 1 1
¿? 1 1 0


1 0 0 1
+1 +1
- 0 0 1 1
0 1 1 0

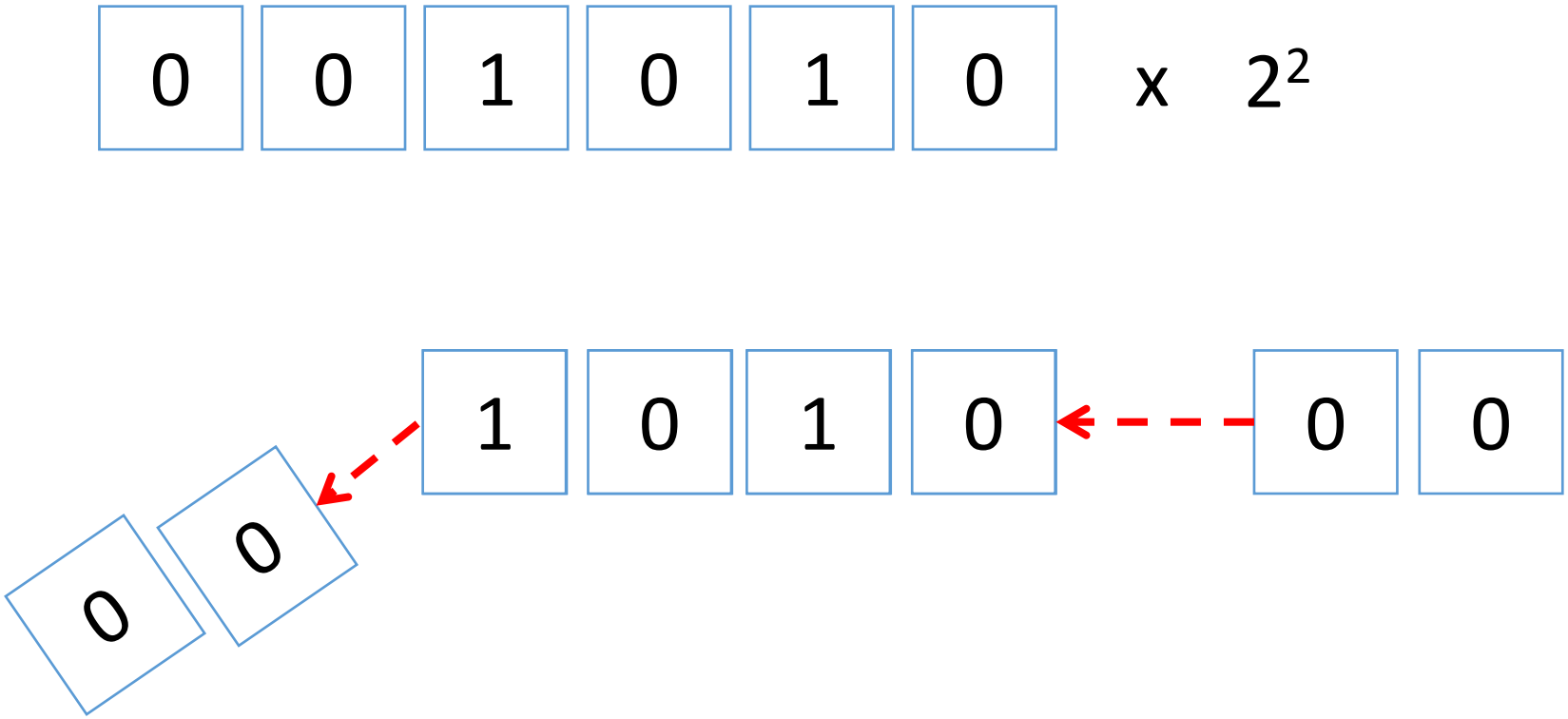
Multiplicar

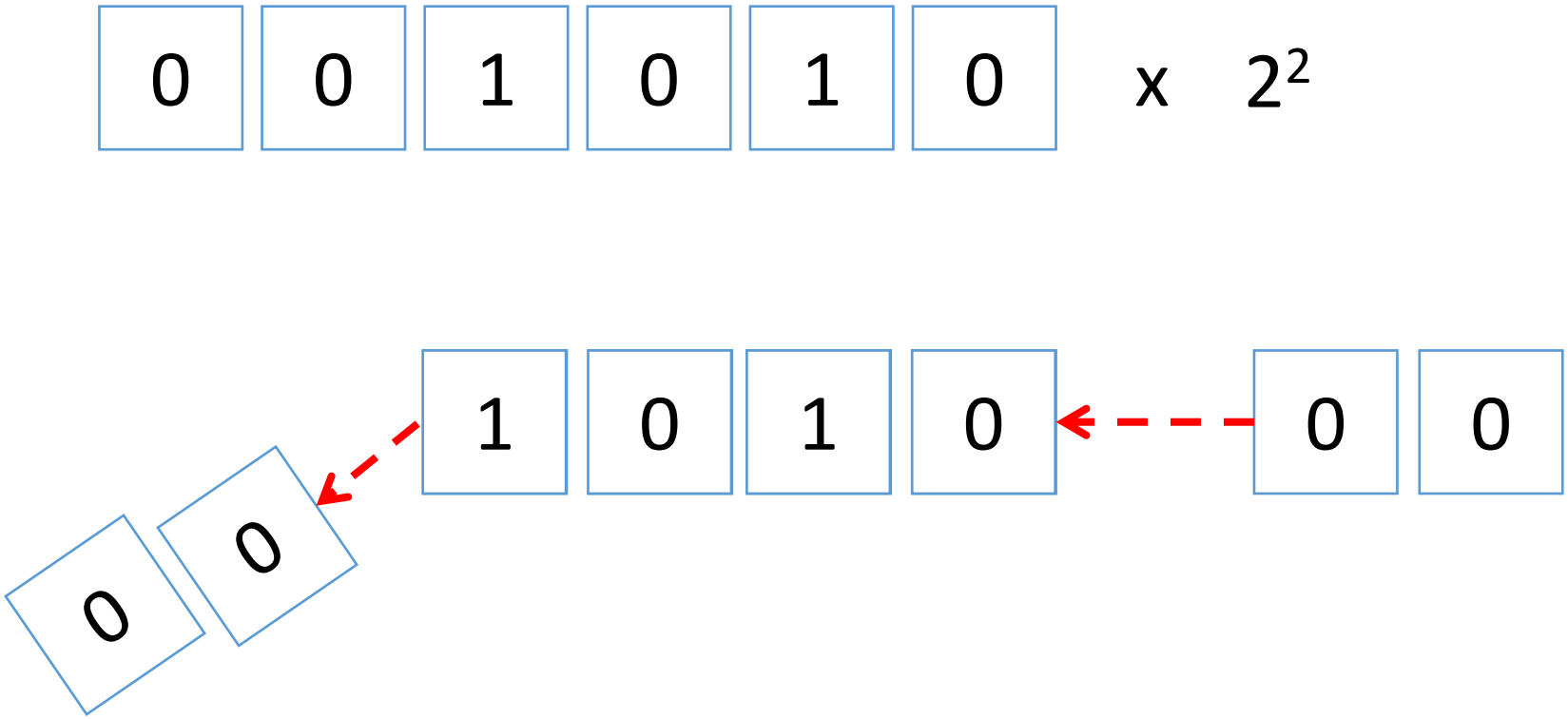
- Multiplicar n veces por la base es equivalente a desplazar n veces los dígitos hacia la izquierda, añadiendo n ceros por la derecha.

- Multiplicar 001010_2 por 2 veces la base:
  - 2 veces la base es multiplicar por $2^2 = 4_{10} \equiv 100_2$

0	0	1	0	1	0

$\times \quad 2^2$


0 0 1 0 1 0
x 2^2
1 0 1 0 0 0
0 0


0 0 1 0 1 0 x 2^2
1 0 1 0 0 0
0 0
1 0 1 0 0 0

- $001010_2 \equiv 10_{10}$
- $101000_2 \equiv 40_{10}$
- $10_{10} x 4_{10} = 40_{10}$

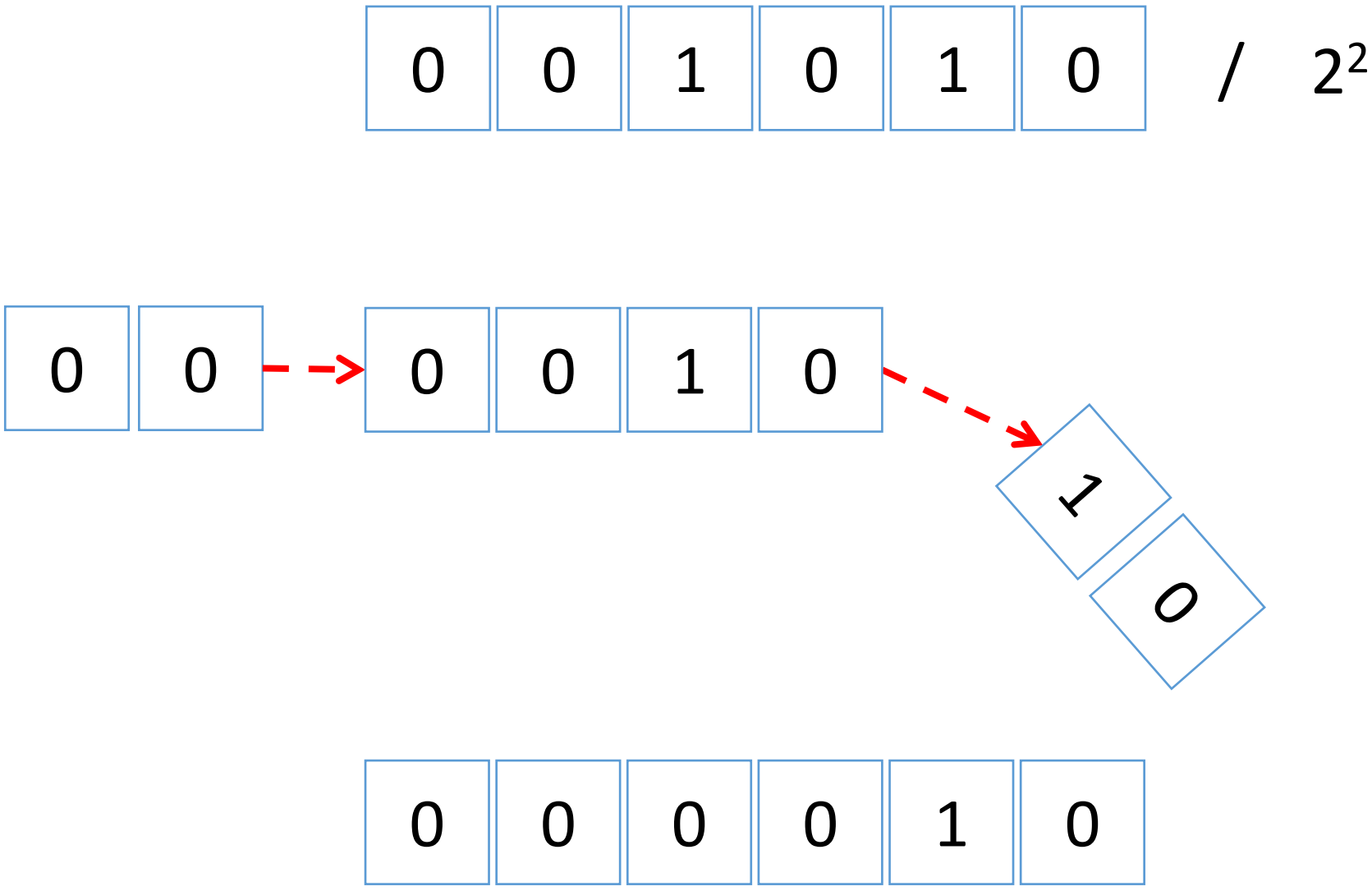
Dividir

- Dividir n veces por la base es equivalente a desplazar n veces los dígitos hacia la derecha, añadiendo n ceros por la izquierda.

- Mediante este proceso se obtiene únicamente el cociente (no el resto).

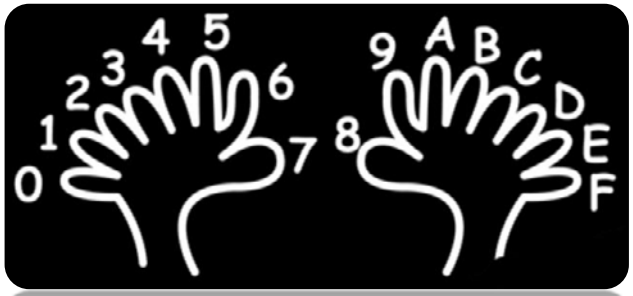
- División entera
  - 7/3 = 2

- Dividir 001010_2 por 2 veces la base:
  - 2 veces la base es dividir por $2^2 = 4_{10} \equiv 100_2$


0 0 1 0 1 0 / 2^2
0 0
0 0 1 0
1
0
0 0 0 0 1 0

- $001010_2 \equiv 10_{10}$
- $000010_2 \equiv 2_{10}$
- $10_{10}/4_{10} = 2_{10}$

El Sistema de numeración hexadecimal

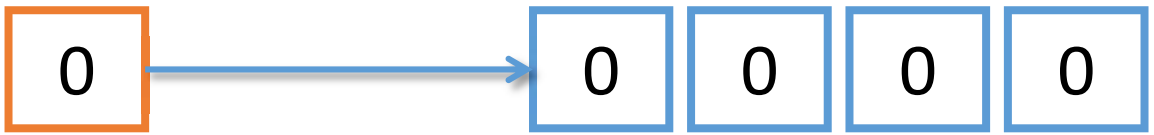


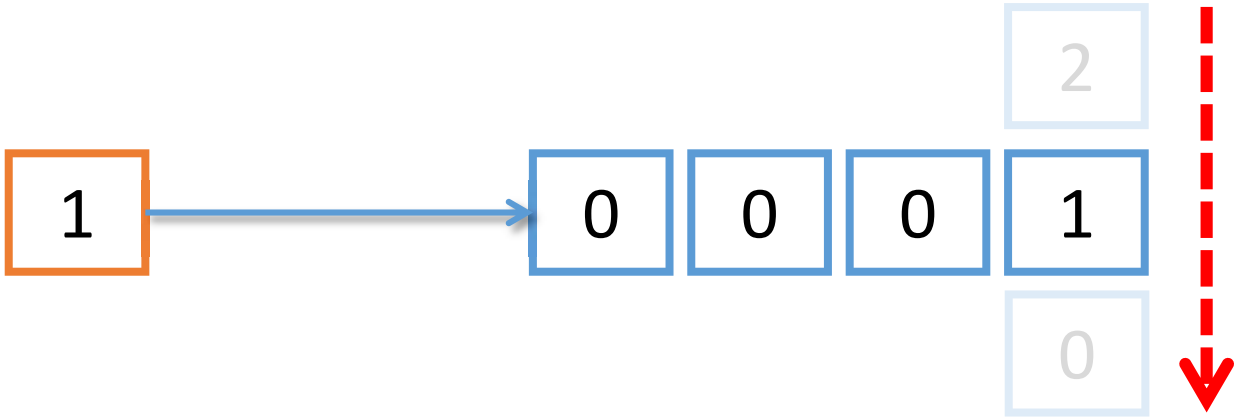
http://spaceplace.nasa.gov/binary-code3/sp/dr-marc-hex-lrg.sp.png

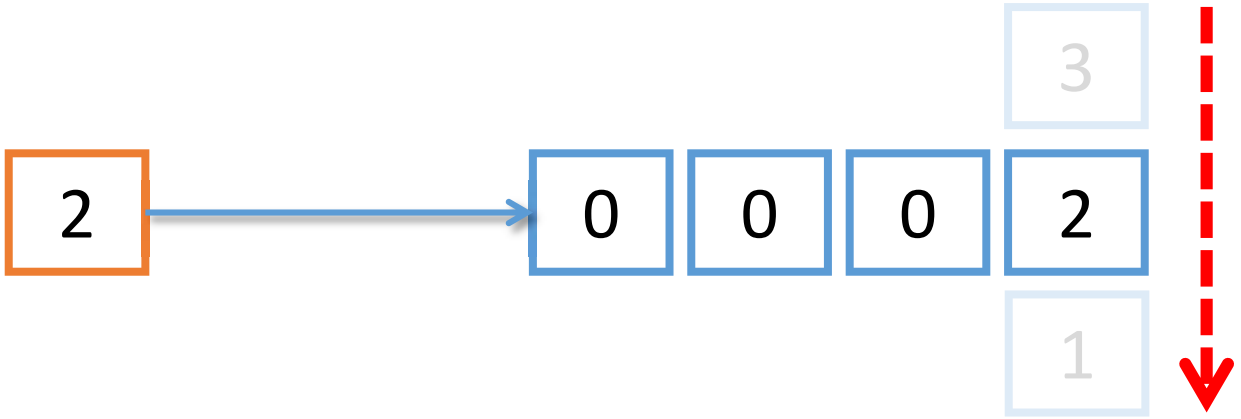
- Es un sistema de numeración **posicional.**

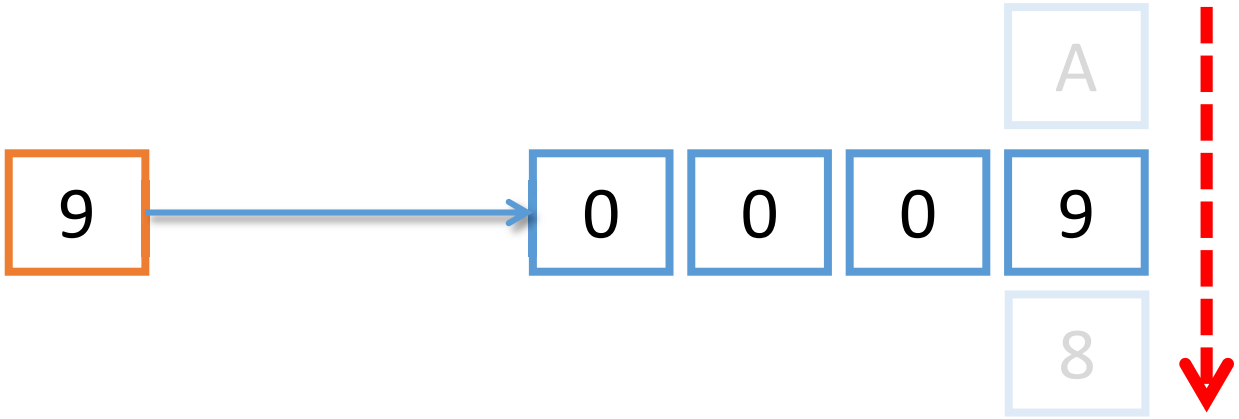
- **16** símbolos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (b=16).

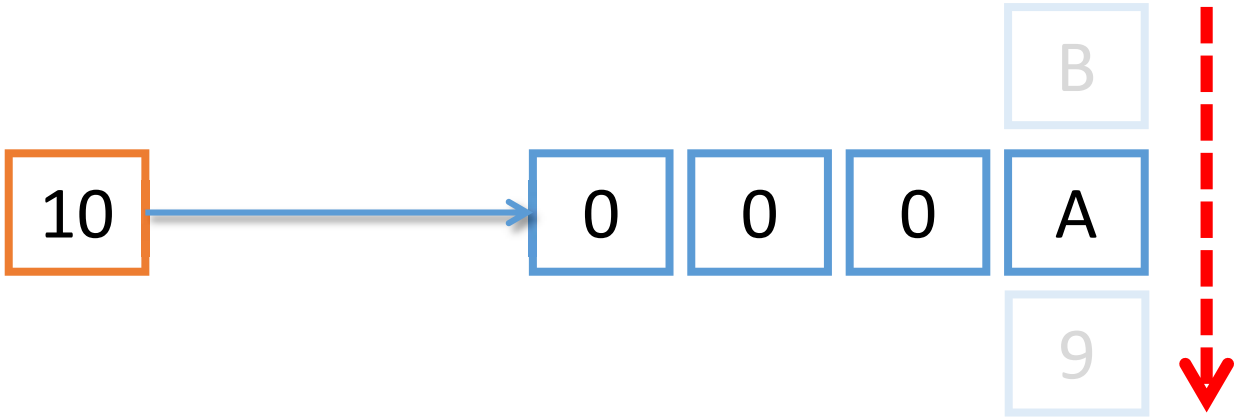
- ¿Cómo funciona el sistema de numeración hexadecimal?
- Supongamos N=4
  - N es el número de dígitos que podemos usar

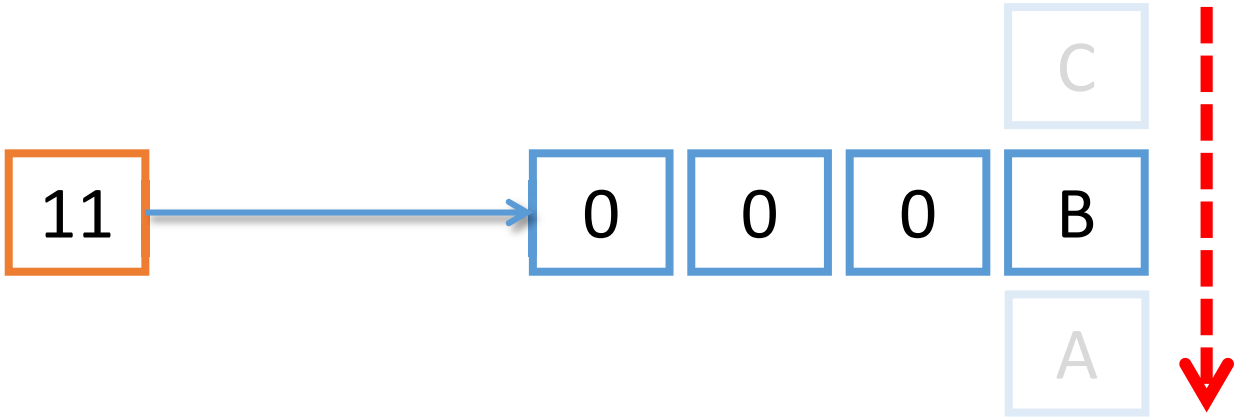


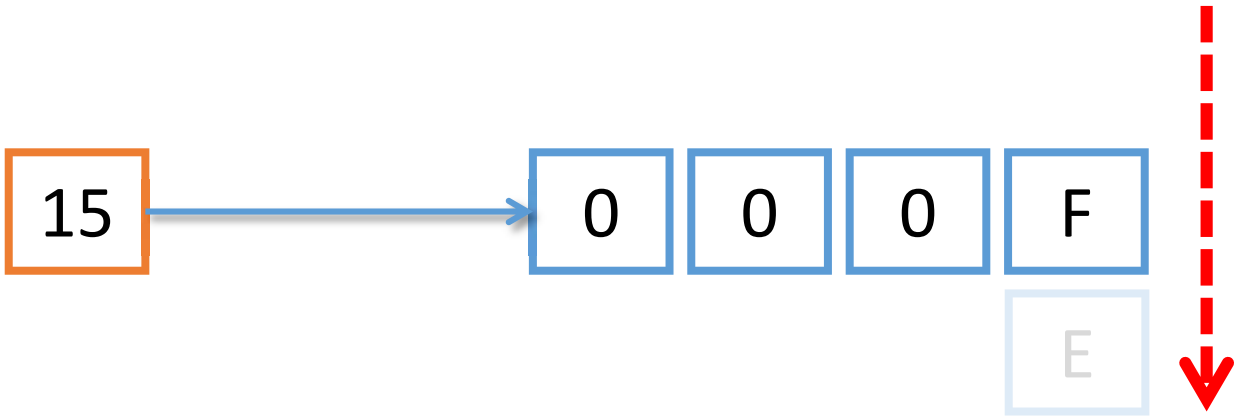

2
1
0
0
0
1
0

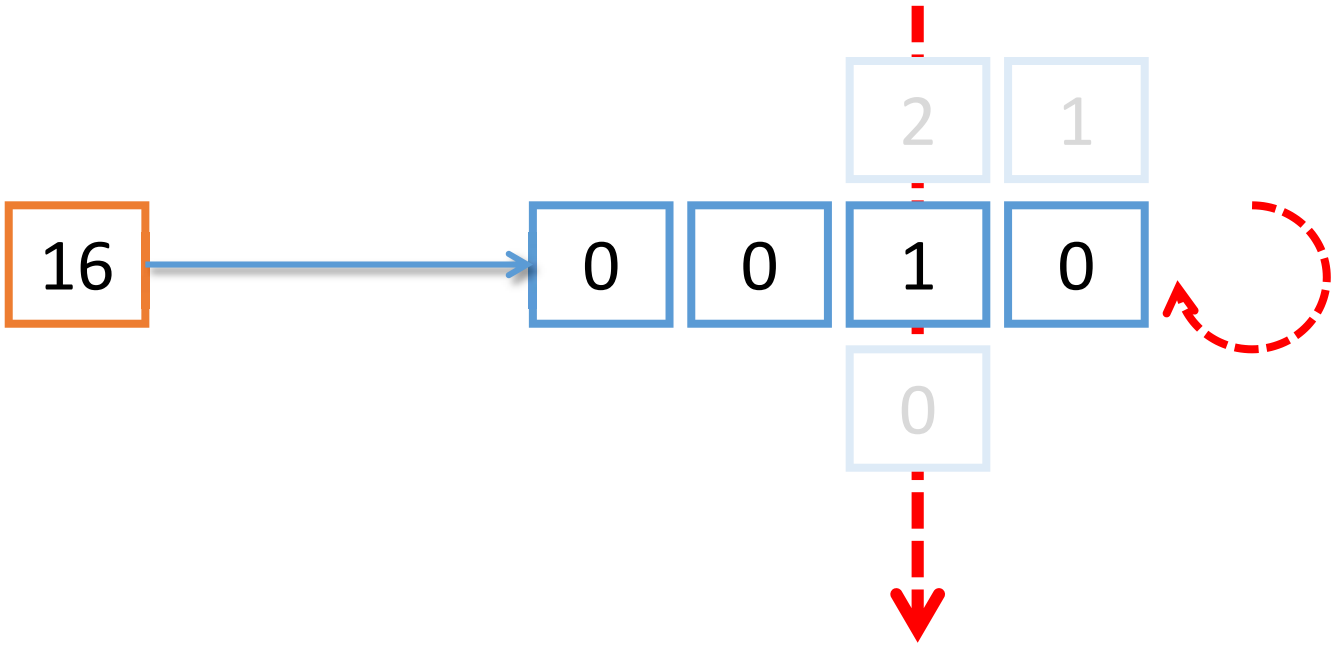

3
2
0
0
0
2
1

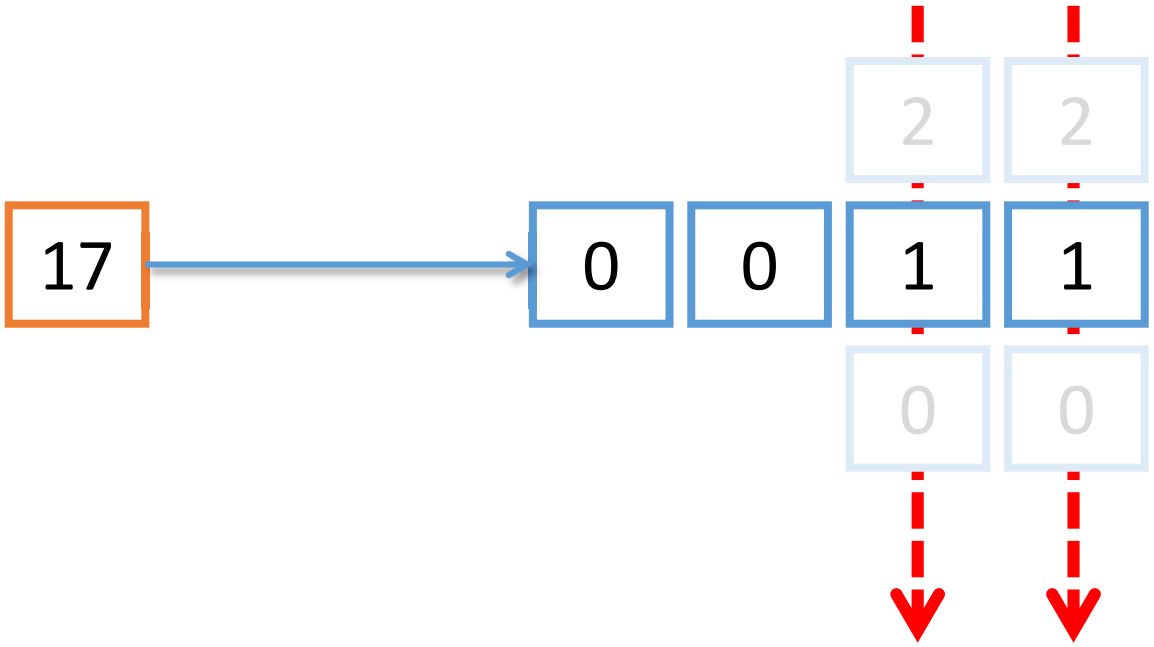

A
9
0
0
0
9
8

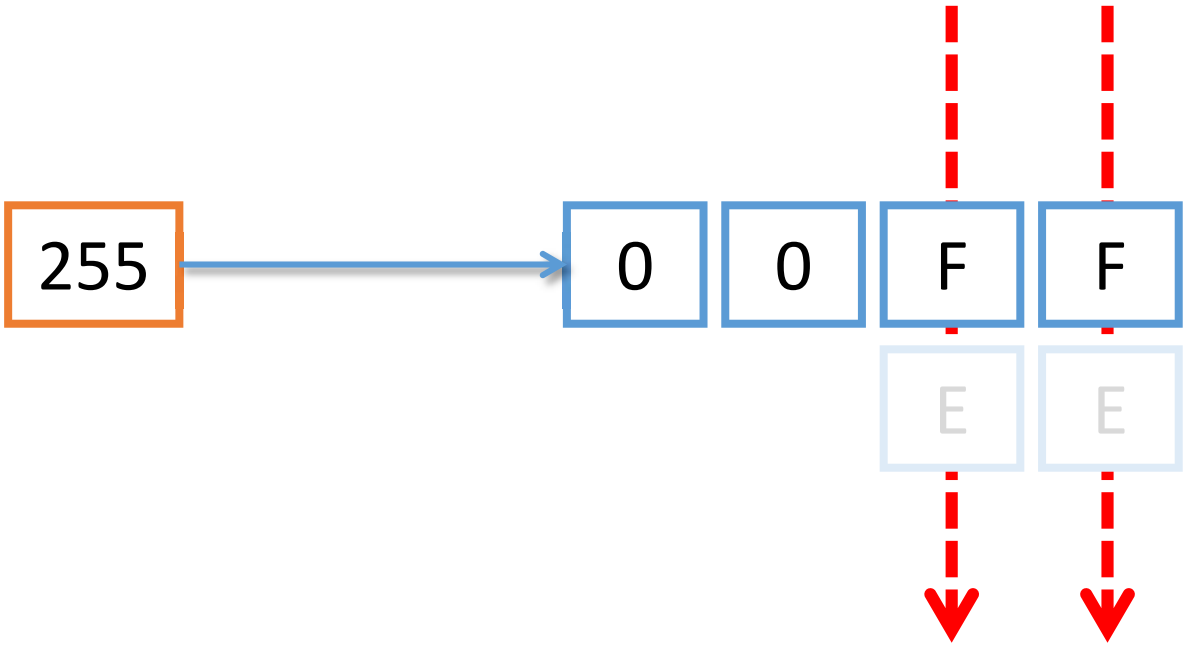

B
10
0
0
0
A
9

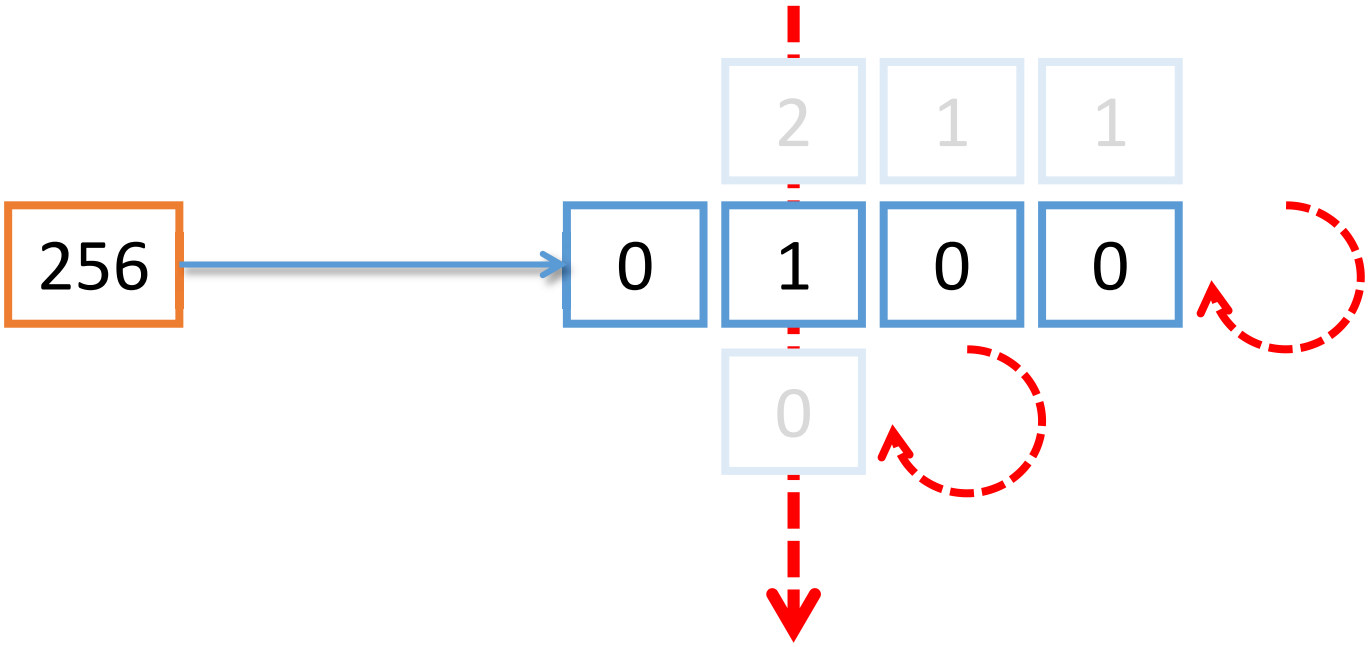

C
11
0
0
0
B
A

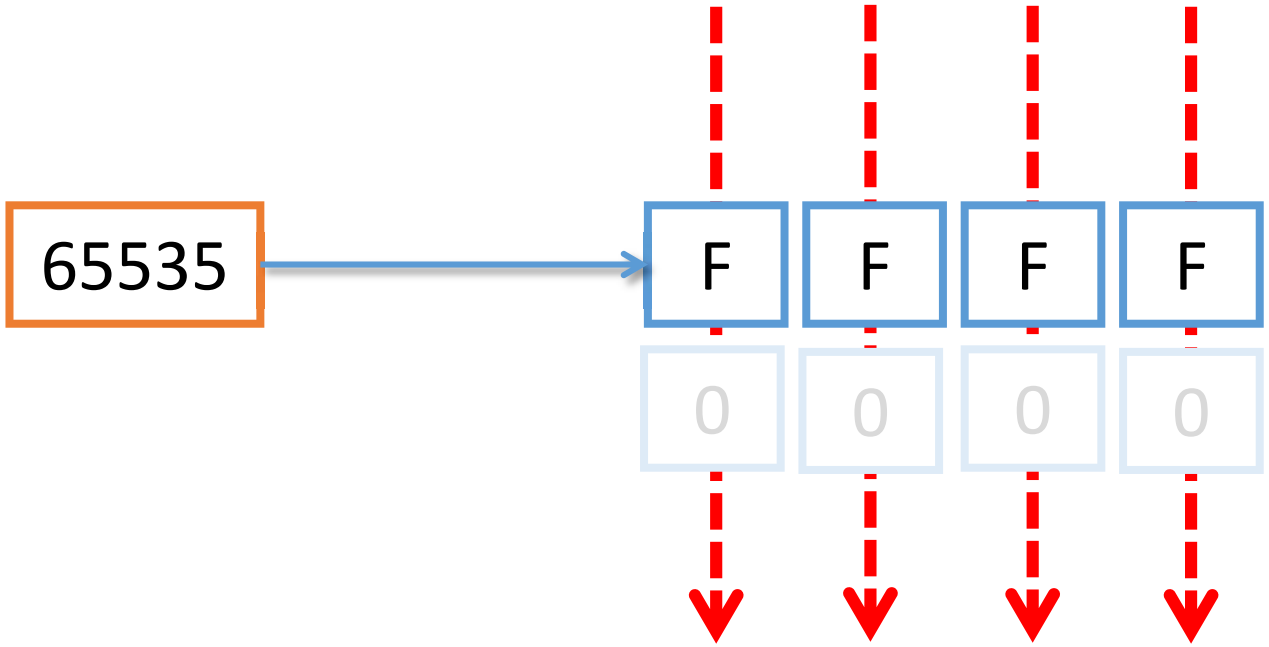

15
0
0
0
F
E


16
0
0
1
0
2
1
0


17
0
0
1
1
2
2
0
0


255
0
0
F
F
E
E


256
0
1
0
0
2
1
1
0


65535
F
F
F
F
0
0
0
0

- Con N dígitos, es posible representar el rango:

$$[\ 0,\ 16^{N} - 1]$$

- Con N=4 dígitos:

$$[0,\ 65535]$$

¿Hexadecimal?

- Relación binario hexadecimal

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

- Calcula el equivalente en hexadecimal del número binario 00101101_2

- Calcula el equivalente en hexadecimal del número binario 00101101_2

- Paso 1: agrupar de 4 en 4:
  - 0010_2 , 1101_2

- Calcula el equivalente en hexadecimal del número binario 00101101_2

- Paso 1: agrupar de 4 en 4:
  - 0010_2 , 1101_2
- Paso 2: Sustituir por dígito hexadecimal
  - $0010_2 \equiv 2_{16}$
  - $1101_2 \equiv D_{16}$

- Calcula el equivalente en hexadecimal del número binario 00101101_2

- Paso 1: agrupar de 4 en 4:
  - 0010_2 , 1101_2
- Paso 2: Sustituir por dígito hexadecimal
  - $0010_2 \equiv 2_{16}$
  - $1101_2 \equiv D_{16}$
- Paso 3: Agrupar dígitos hexadecimales:
  - $00101101_2 \equiv 2D_{16}$

- Calcula el equivalente en hexadecimal del número binario 1001101011_2

- Calcula el equivalente en hexadecimal del número binario 1001101011_2

- **ATENCIÓN:**
  - El número de dígitos no es múltiplo de 4.
  - Solución: añadir ceros por la izquierda

- Calcula el equivalente en hexadecimal del número binario 1001101011_2

- **ATENCIÓN:**
  - El número de dígitos no es múltiplo de 4.
  - Solución: añadir ceros por la izquierda

- $1001101011_2 = 001001101011_2$

- Calcula el equivalente en hexadecimal del número binario 1001101011_2

- **ATENCIÓN:**
  - El número de dígitos no es múltiplo de 4.
  - Solución: añadir ceros por la izquierda

- $1001101011_2 = 001001101011_2$
- $0010_2\ 0110_2\ 1011_2$

- Calcula el equivalente en hexadecimal del número binario 1001101011_2

- **ATENCIÓN:**
  - El número de dígitos no es múltiplo de 4.
  - Solución: añadir ceros por la izquierda

- $1001101011_2 = 001001101011_2$
- $0010_2\ 0110_2\ 1011_2$

- Calcula el equivalente en hexadecimal del número binario 1001101011_2

- **ATENCIÓN:**
  - El número de dígitos no es múltiplo de 4.
  - Solución: añadir ceros por la izquierda

- $1001101011_2 = 001001101011_2$
- $0010_2\ 0110_2\ 1011_2$
- $2_{16}\ 6_{16}\ B_{16}$

- Calcula el equivalente en hexadecimal del número binario 1001101011_2

- **ATENCIÓN:**
  - El número de dígitos no es múltiplo de 4.
  - Solución: añadir ceros por la izquierda

- $1001101011_2 = 001001101011_2$
- $0010_2\ 0110_2\ 1011_2$
- $2_{16}\ 6_{16}\ B_{16}$

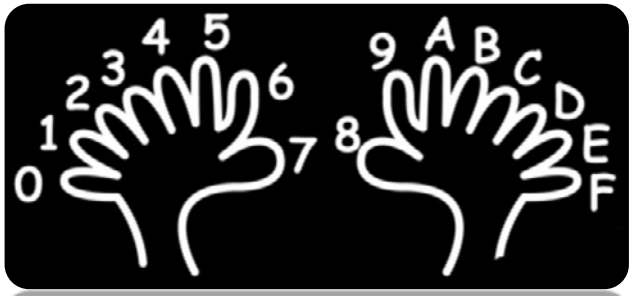
- $1001101011_2 \equiv 26B_{16}$

- Resumen:
  - Un número de N dígitos en binario se expresa en N/4 dígitos en hexadecimal

- Los números en hexadecimal se escriben:
  - Subíndice 16
  - Prefijo 0x

$$0x26B \equiv 26B_{16}$$

El Sistema de numeración hexadecimal



http://spaceplace.nasa.gov/binary-code3/sp/dr-marc-hex-lrg.sp.png

De Hexadecimal a Decimal

- Dos métodos:
  - Aplicar la fórmula con b=16
  - Convertir primero a binario y luego aplicar la fórmula con b=2

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Convierte $AE4_{16}$ a decimal

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Convierte $AE4_{16}$ a decimal:
  - $A_{16} \equiv 10_{10}$
  - $E_{16} \equiv 14_{10}$
  - $4_{16} \equiv 4_{10}$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Convierte $AE4_{16}$ a decimal:
  - $A_{16} \equiv 10_{10}$
  - $E_{16} \equiv 14_{10}$
  - $4_{16} \equiv 4_{10}$

$$AE4_{16} = 10*16^2 + 14*16^1 + 4*16^0$$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Convierte $AE4_{16}$ a decimal:
  - $A_{16} \equiv 10_{10}$
  - $E_{16} \equiv 14_{10}$
  - $4_{16} \equiv 4_{10}$

$AE4_{16} = 10*16^2 + 14*16^1 + 4*16^0$

$= \quad 10*256 + 14*16 + 4*1$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Convierte $AE4_{16}$ a decimal:
  - $A_{16} \equiv 10_{10}$
  - $E_{16} \equiv 14_{10}$
  - $4_{16} \equiv 4_{10}$

$AE4_{16} = 10*16^2 + 14*16^1 + 4*16^0$

$= 10*256 + 14*16 + 4*1$

$= 2560 + 224 + 4$

$= 2788$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Convierte $AE4_{16}$ a decimal
  - $A_{16} \equiv 1010_2$
  - $E_{16} \equiv 1110_2$
  - $4_{16} \equiv 0100_2$

$$AE4_{16} \equiv 101011100100_2$$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Convierte $AE4_{16}$ a decimal
  - $A_{16} \equiv 1010_2$
  - $E_{16} \equiv 1110_2$
  - $4_{16} \equiv 0100_2$

$$AE4_{16} \equiv 101011100100_2$$

$$101011100100_2 = 2^{11} + 2^9 + 2^7 + 2^6 + 2^5 + 2^2$$

$$x_{10} = \sum_{p=N-1}^{0} x_b[p] * b^p$$

- Convierte $AE4_{16}$ a decimal
  - $A_{16} \equiv 1010_2$
  - $E_{16} \equiv 1110_2$
  - $4_{16} \equiv 0100_2$

$AE4_{16} \equiv 101011100100_2$

$101011100100_2 = 2^{11} + 2^9 + 2^7 + 2^6 + 2^5 + 2^2$

$= 2048 + 512 + 128 + 64 + 32 + 4$

$= 2788$

De Decimal a Hexadecimal

- Dos métodos:
  - Divisiones sucesivas por 16
  - Convertir primero a binario y luego agrupar de 4 en 4

- Convertir 46_{10} a hexadecimal:

46	16
14	2

¿?	¿?

- Convertir 46_{10} a hexadecimal:

- Convertir 46_{10} a hexadecimal:

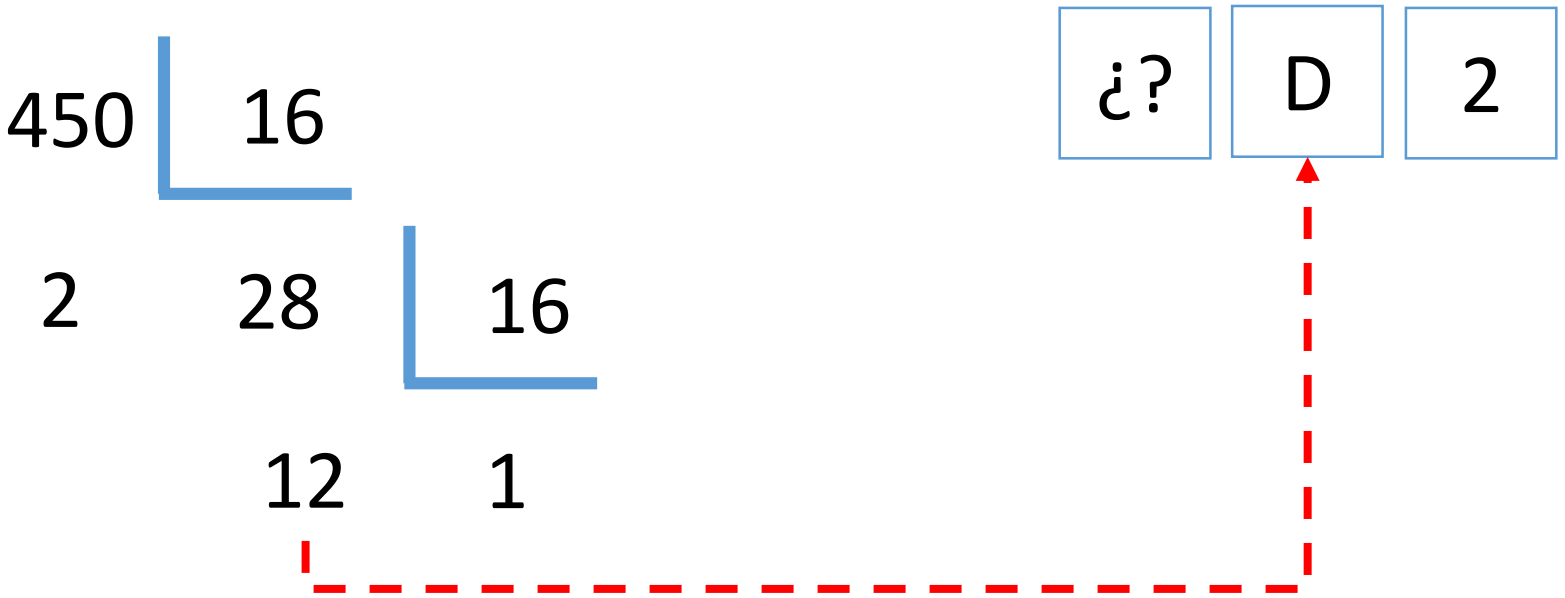
- Convertir 450_{10} a hexadecimal:

450	16
2	28

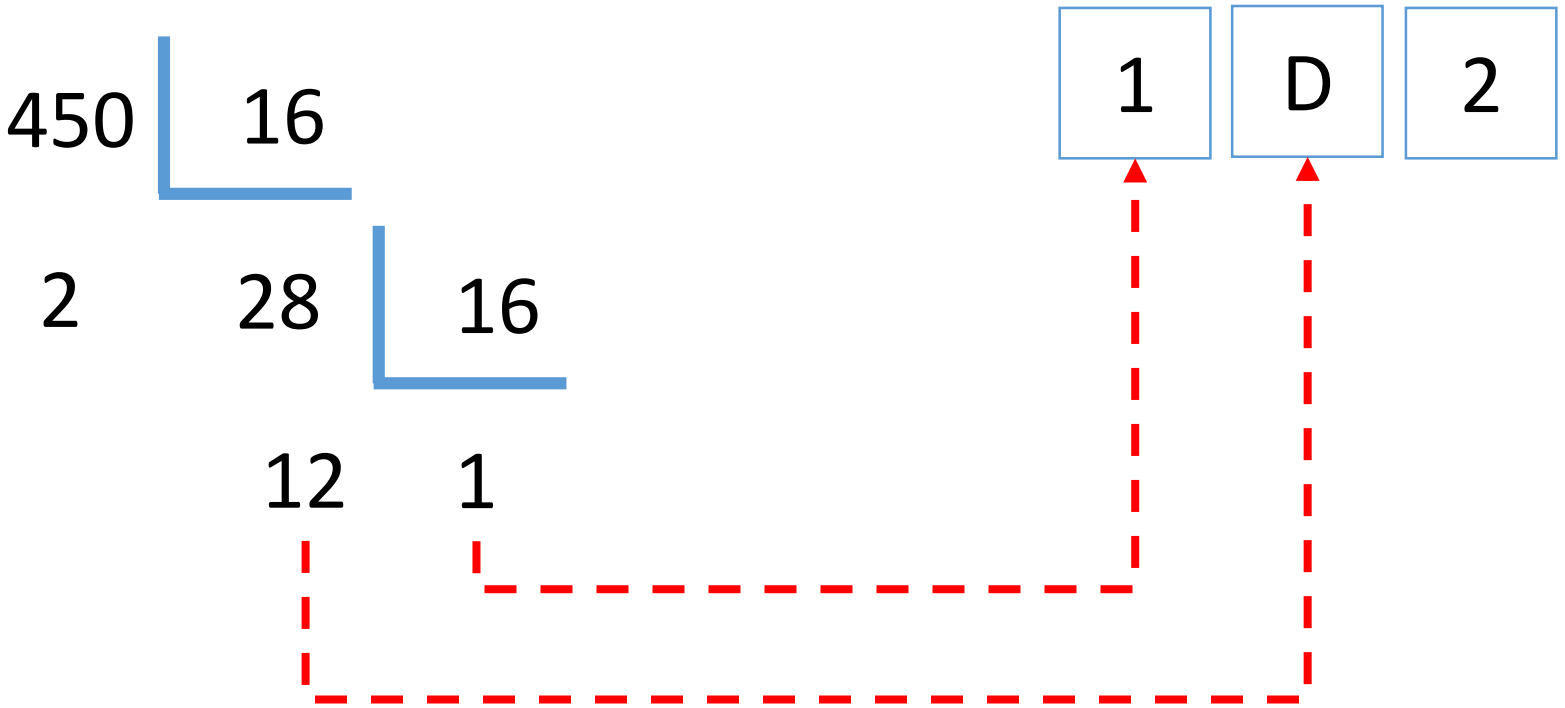
¿?	¿?	¿?

- Convertir 450_{10} a hexadecimal:

- Convertir 450_{10} a hexadecimal:



- Convertir 450_{10} a hexadecimal:



- Convertir 46_{10} a hexadecimal:

$46_{10} \equiv 101110_2$

- Convertir 46_{10} a hexadecimal:

$46_{10} \equiv 101110_2$

$101110_2 = 00101110_2$

- Convertir 46_{10} a hexadecimal:

$46_{10} \equiv 101110_2$

$101110_2 = 00101110_2$

$0010_2 \equiv 2_{16}$
$1110_2 \equiv E_{16}$

$46_{10} \equiv 101110_2 \equiv 2E_{16}$

- Convertir 446_{10} a hexadecimal:

$446_{10} \equiv 110111110_2$

- Convertir 446_{10} a hexadecimal:

$446_{10} \equiv 110111110_2$

$110111110_2 = 000110111110_2$

- Convertir 446_{10} a hexadecimal:

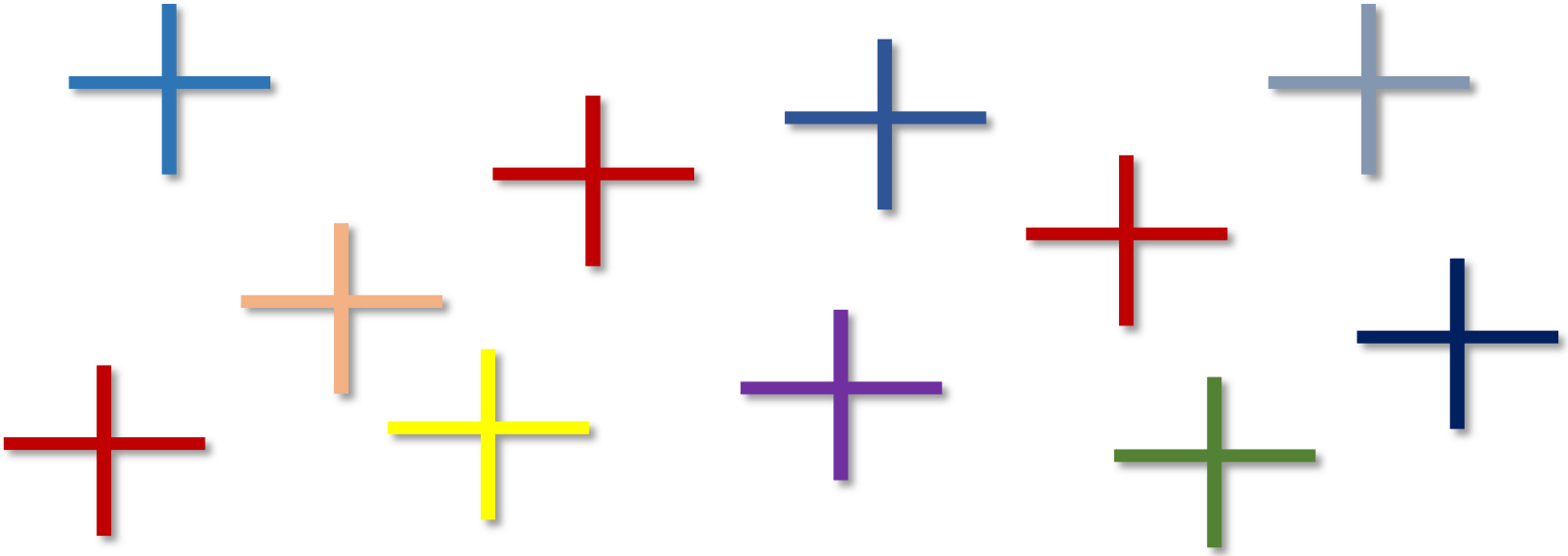
$446_{10} \equiv 110111110_2$

$110111110_2 = 000110111110_2$

$0001_2 \equiv 1_{16;}$
$1011_2 \equiv B_{16}$
$1110_2 \equiv E_{16}$

$46_{10} \equiv 110111110_2 \equiv 1BE_{16}$

Codificación de números enteros sin signo



Nota

Para facilitar la lectura de números en binario usaremos un pequeño espacio cada 4 dígitos.

$1000\ 0001\ 0101_2$

$10\ 0101\ 0101_2$

Programas informáticos y lenguajes de programación

Programas y Lenguajes de programación

Programa Informático

http://pixabay.com/es/monitor-pantalla-equipo-149362/

Programas y Lenguajes de programación

Los programas informáticos se escriben usando Lenguajes de programación.

- Lenguajes de programación comunes:
  - C, C++
  - Python
  - Java
  - Javascript

Tipos de datos

Tipos de datos

Para poder realizar programas es necesario conocer los tipos de datos existentes en ese lenguaje.

Tipos de datos

Preguntas:

- ¿Cuál es el rango de valores que necesitamos representar?

Tipos de datos

- ¿Cuántos artículos diferentes vamos a tener?
- ¿Qué stock máximo podemos tener de cada artículo?
- ¿Cuántas ventas podemos llegar a tener de cada artículo?

http://pixabay.com/es/sudaderas-su%C3%A9teres-exposici%C3%B3n-428607/

Tipos de datos

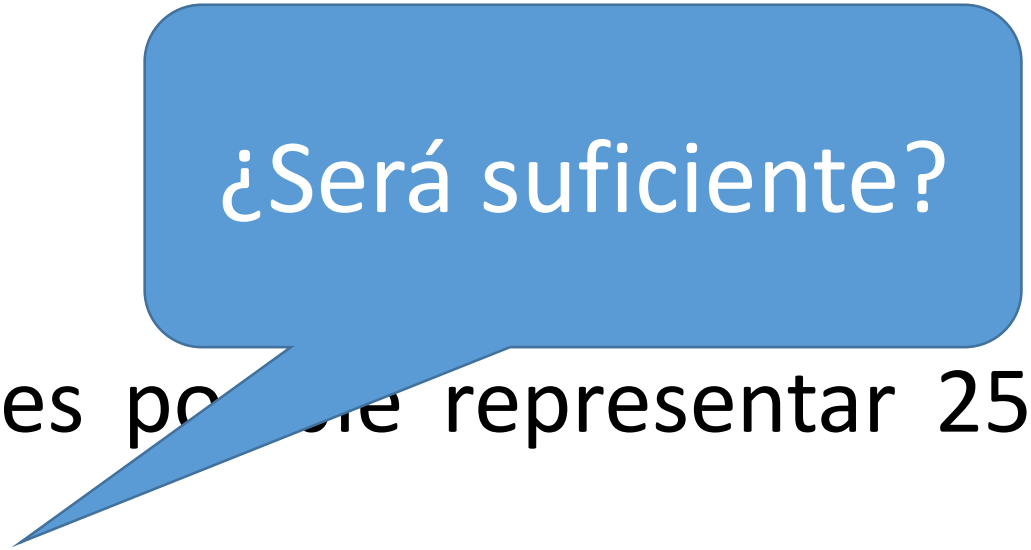
- En los tres casos es un número positivo.
- ¿Cómo de grande?

Tipos de datos

- 1 byte.
- Con 1 byte es posible representar 256 valores.
- El más pequeño 0, el más grande 255.

Tipos de datos

- 1 byte.
- Con 1 byte es posible representar 256 valores.
- El más pequeño 0, el más grande 255.



Tipos de datos

- ¿Cuántos artículos diferentes vamos a tener?

Tipos de datos

- ¿Cuántos artículos diferentes vamos a tener?

- ¿Qué stock máximo podemos tener de cada artículo?

Tipos de datos

- ¿Cuántos artículos diferentes vamos a tener?
- ¿Qué stock máximo podemos tener de cada artículo?
- ¿Cuántas ventas podemos llegar a tener de cada artículo?

Tipos de datos más comunes

Tipos de datos

- Tipos de datos más comunes en Java/C:

Nombre	Número de bits	Dígitos en Hex	Rango
byte	8	2	[0, 255]
short	16	4	[0, 65 535]
int	32	8	[0, 4 294 967 295]
long	64	16	[0, 18 446 744 073 709 551 615]

El tipo *byte*

El tipo *byte*

- El tipo *byte* usa 8 bits
- Rango [0, 255]
- Al expresar un número usando el tipo *byte*, se deberá usar 8 bits.
  - Rellenar con 0s por la izquierda, si es necesario.

El tipo *byte*

- Expresa 58_{10} en binario usando el tipo de datos *byte*.

El tipo *byte*

- Expresa 58_{10} en binario usando el tipo de datos *byte*.

- $58_{10} \equiv 11\ 1010_2$

El tipo *byte*

- Expresa 58_{10} tipo de datos *byte*.

- $58_{10} \equiv 11\ 1010_2$

Espacio cada cuatro dígitos binarios

El tipo *byte*

- Expresa 58_{10} en binario usando el tipo de datos *byte*.
- $58_{10} \equiv 11\ 1010_2$
- Con el tipo *byte*:
  - $58_{10} \equiv 0011\ 1010_2 \equiv 3A_{16}$

El tipo *byte*

- Expresa 43_{10} en binario usando el tipo de datos *byte*.

El tipo *byte*

- Expresa 43_{10} en binario usando el tipo de datos *byte*.

- $43_{10} \equiv 10\ 1011_2$

El tipo *byte*

- Expresa 43_{10} en binario usando el tipo de datos *byte*.
- $43_{10} \equiv 10\ 1011_2$
- Con el tipo *byte*:
  - $43_{10} \equiv 0010\ 1011_2 \equiv 2B_{16}$

El tipo *byte*

- Expresa 12_{10} en binario usando el tipo de datos *byte*.

El tipo *byte*

- Expresa 12_{10} en binario usando el tipo de datos *byte*.

- $12_{10} \equiv 1100_2$

El tipo *byte*

- Expresa 12_{10} en binario usando el tipo de datos *byte*.
- $12_{10} \equiv 1100_2$
- Con el tipo *byte*:
  - $12_{10} \equiv 0000\ 1100_2 \equiv 0C_{16}$

El tipo *byte*

- Expresa 276_{10} en binario usando el tipo de datos *byte*.

El tipo *byte*

- Expresa 276_{10} en binario usando el tipo de datos *byte*.
- No es posible representarlo.

El tipo *short*

El tipo *short*

- El tipo *short* usa 16 bits.
- Rango [0, 65 535]

El tipo *short*

- Expresa 58_{10} en binario usando el tipo de datos *short*.

El tipo *short*

- Expresa 58_{10} en binario usando el tipo de datos *short*.

- $58_{10} \equiv 11\ 1010_2$

El tipo *short*

- Expresa 58_{10} en binario usando el tipo de datos *short*.
- $58_{10} \equiv 11\ 1010_2$
- Con el tipo *short*:
  - $58_{10} \equiv 0000\ 0000\ 0011\ 1010_2 \equiv 003A_{16}$

El tipo *short*

- Con byte:
  - $58_{10} \equiv 0011\ 1010_2 \equiv 3A_{16}$
- Con short:
  - $58_{10} \equiv 0000\ 0000\ 0011\ 1010_2 \equiv 003A_{16}$

El tipo *short*

- Expresa 2358_{10} en binario usando el tipo de datos *short*.

El tipo *short*

- Expresa 2358_{10} en binario usando el tipo de datos *short*.

- $2358_{10} \equiv 1001\ 0011\ 0110_2$

El tipo *short*

- Expresa 2358_{10} en binario usando el tipo de datos *short*.
- $2358_{10} \equiv 1001\ 0011\ 0110_2$
- Con el tipo *short*:
  - $2358_{10} \equiv 0000\ 1001\ 0011\ 0110_2 \equiv 0936_{16}$

El tipo *int*

El tipo *int*

- El tipo *int* usa 32 bits.
- Rango [0, 4 294 967 295]

El tipo *int*

- Expresa 2358_{10} en binario usando el tipo de datos *int*.

El tipo *int*

- Expresa 2358_{10} en binario usando el tipo de datos *int*.

- $2358_{10} \equiv 1001\ 0011\ 0110_2$

El tipo *int*

- Expresa 2358_{10} en binario usando el tipo de datos *int*.

- $2358_{10} \equiv 1001\ 0011\ 0110_2$

- Con el tipo *int*:
  - $2358_{10} \equiv 0000\ 0000\ 0000\ 0000\ 0000\ 1001\ 0011\ 0110_2 \equiv 00000936_{16}$

El tipo *int*

- Expresa $451\ 358_{10}$ en binario usando el tipo de datos *int*.

El tipo *int*

- Expresa $451\ 358_{10}$ en binario usando el tipo de datos *int*.

- $451\ 358_{10} \equiv 110\ 1110\ 0011\ 0001\ 1110_2$

El tipo *int*

- Expresa $451\ 358_{10}$ en binario usando el tipo de datos *int*.

- $451\ 358_{10} \equiv 110\ 1110\ 0011\ 0001\ 1110_2$

- Con el tipo *int*:
  - $451\ 358_{10} \equiv 0000\ 0000\ 0000\ 0110\ 1110\ 0011\ 0001\ 1110_2 \equiv 0006E31E_{16}$

El tipo *long*

El tipo *long*

- El tipo *long* usa 64 bits.
- Rango [0, 18 446 744 073 709 551 615]

El tipo *long*

- Expresa $451\ 358_{10}$ en binario usando el tipo de datos *long*.

El tipo *long*

- Expresa $451\ 358_{10}$ en binario usando el tipo de datos *long*.

- $451\ 358_{10} \equiv 110\ 1110\ 0011\ 0001\ 1110_2$

El tipo *long*

- Expresa $451\ 358_{10}$ en binario usando el tipo de datos *long*.

- $451\ 358_{10} \equiv 110\ 1110\ 0011\ 0001\ 1110_2$

- Con el tipo *long*:
  - $451\ 358_{10} \equiv 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0110\ 1110\ 0011\ 0001\ 1110_2 \equiv 00000000006E31E_{16}$

Elección de tipos y memoria

Tipos de datos y memoria

- ¿Cuántos artículos diferentes vamos a tener?
- ¿Qué stock máximo podemos tener de cada artículo?
- ¿Cuántas ventas podemos llegar a tener de cada artículo?

http://pixabay.com/es/sudaderas-su%C3%A9teres-exposici%C3%B3n-428607/

Tipos de datos y memoria

- Supongamos:
  - Artículos diferentes: 1000
  - Stock máximo de cada artículo: 55 000
  - Número máximo de ventas esperadas de cada artículo: 5 000 000 000

Tipos de datos y memoria

- Opción 1:
  - Elegimos *byte*.
  - Se necesita poca memoria
- No es posible

Tipos de datos y memoria

- Opción 2:
  - Elegimos *long*.
  - Se necesita mucha memoria

Memoria necesaria:

2 * 1000 * 64 = 128 000 bits =

= 128 000 / 8 bytes = 16 000 bytes = 16kB

Tipos de datos y memoria

- Opción 3:
  - Elegimos *short* para el stock y *long* para el número de ventas.

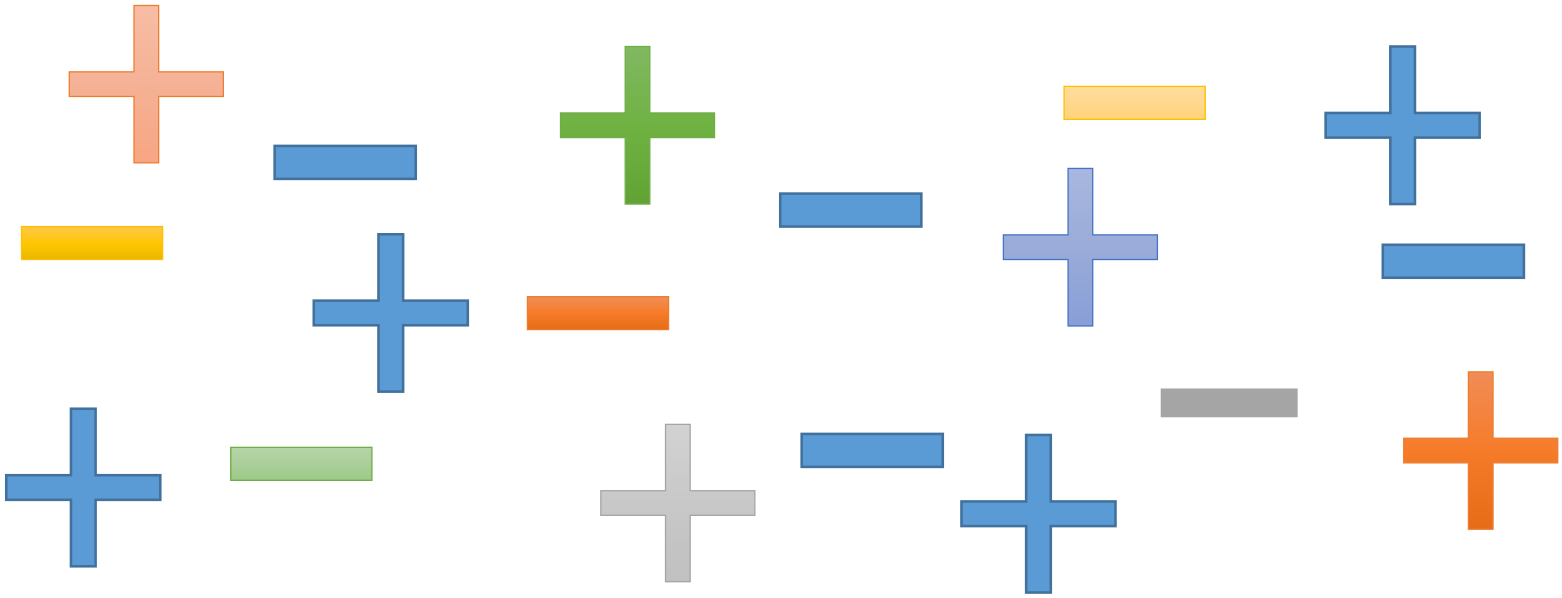
Memoria necesaria:

1000 * 16 + 1000* 64 = 80 000 bits =

80 000 / 8 bytes = 10 000 bytes = 10kB

Casi 40% de ahorro

Codificación de números enteros con signo



Números con signo

$$
\begin{array}{r}
47 \\
-\ 56 \\
\hline
-9
\end{array}
$$

Números con signo

- En decimal se usa el símbolo **“-”**
- En binario **no podemos** usar el símbolo: “-” para expresar cantidades negativas.
- Los número negativos se expresan en binario usando los símbolos **“0” y “1”.**

Números con signo

Signo-Magnitud

Exceso Z

Complemento a 2

Números con signo

Signo-Magnitud

Exceso Z

Complemento a 2

Muy fácil de entender

Números con signo

Signo-Magnitud

Exceso Z

Complemento a 2

El que realmente se usa

Números con signo

Signo-Magnitud

Exceso Z

Complemento a 2

Se usa en el método de representación de números reales

Números con signo

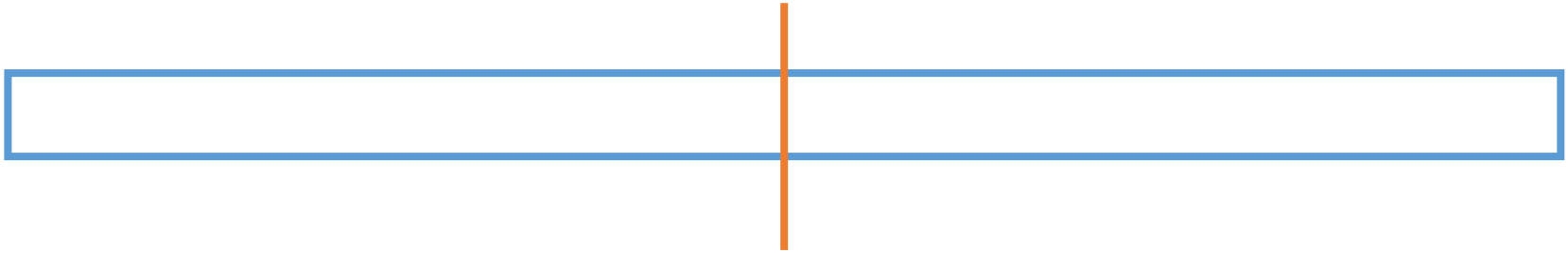
- Con el mismo número de bits el número positivo máximo se reduce **a la mitad**.

Números con signo

- Con el mismo número de bits el número positivo máximo se reduce a la mitad.

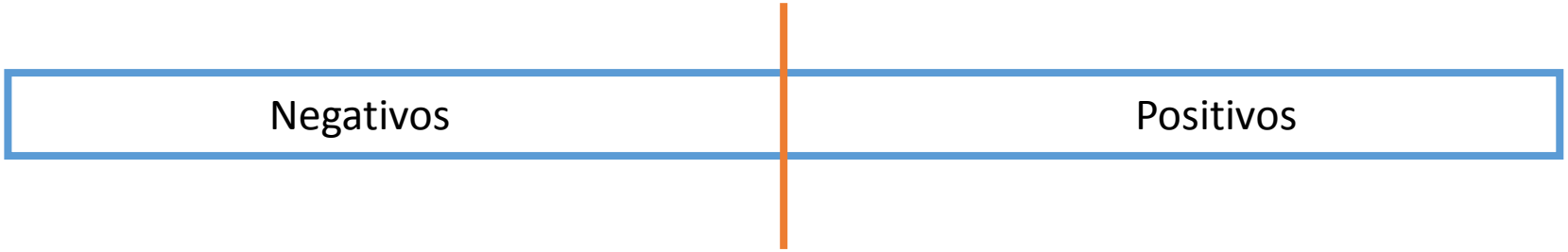
Números con signo

- Con el mismo número de bits el número positivo máximo se reduce a la mitad.



Números con signo

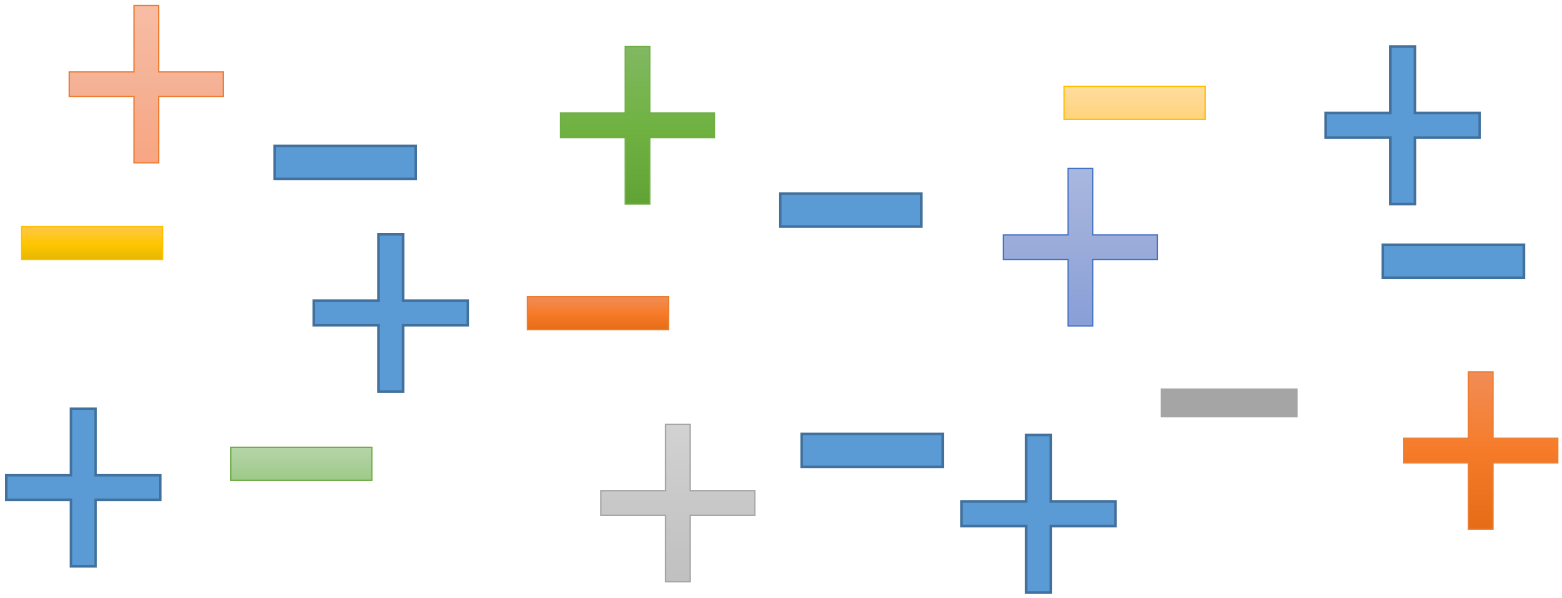
- Con el mismo número de bits el número positivo máximo se reduce a la mitad.



Números con signo

- Operaciones con signo y sin signo:
  - Un **mismo conjunto de bits** puede expresar un número u otro según como se informe al **procesador** sobre si la operación es **con signo** o **sin signo**.

Codificación de números enteros con signo



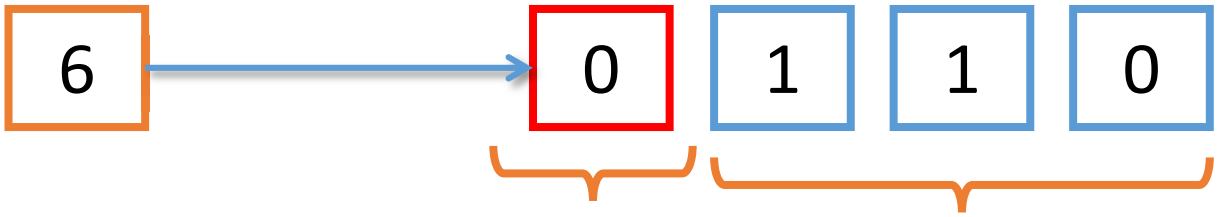
Signo-Magnitud

Signo-Magnitud

- N dígitos:
  - 1 para el signo (el más significativo):
    - “0” para positivos
    - “1” para negativos
  - N-1 para la cantidad:
    - Valor absoluto

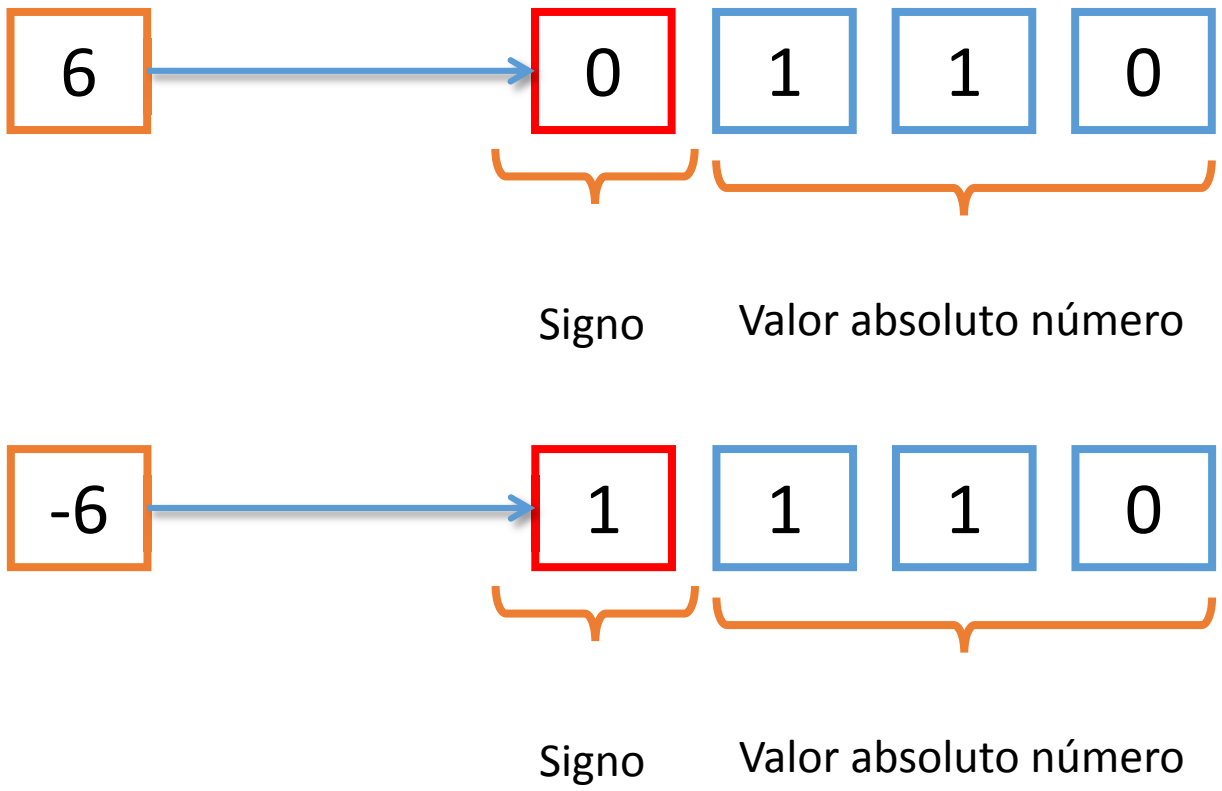
Signo-Magnitud

- N =4



Signo-Magnitud

- N =4



Signo-Magnitud

- Los número **x** y **–x** solo se diferencian en el primer bit (el de más a la izquierda):

  - $6_{10} \equiv 0110_2$
  - $-6_{10} \equiv 1110_2$

Signo-Magnitud

- Rango:

$$[\ -2^{N-1}-1,\ +2^{N-1}\ -1]$$

Signo-Magnitud

- Con N=4, **sin signo**

$[0, +2^N-1]$ $[0, 15]$

- Con N=4, **con signo**, en Signo-Magnitud

$[-2^{N-1}-1, +2^{N-1} -1]$ $[-7, +7]$

Signo-Magnitud

- Con N=8, **sin signo**

$[0, +2^N-1]$ $[0, 255]$

- Con N=8, **con signo**, en Signo-Magnitud

$[-2^{N-1}-1, +2^{N-1} -1]$ $[-127, +127]$

Signo-Magnitud

- Expresa el número -31_{10} en Signo-Magnitud con N=8:

Signo-Magnitud

- Expresa el número -31_{10} en Signo-Magnitud con N=8:
  - Es negativo -> primer bit “1”
  - $|-31_{10}| = 31_{10}$ con 7 bits ≡ $001\ 1111_2$

Signo-Magnitud

- Expresa el número -31_{10} en Signo-Magnitud con N=8:
  - Es negativo -> primer bit “1”
  - $|-31_{10}| = 31_{10}$ con 7 bits ≡ $001\ 1111_2$
- Resultado:
  - $-31_{10} \equiv 1001\ 1111_2$

Signo-Magnitud

- Expresa el número 72_{10} en Signo-Magnitud con N=8:

Signo-Magnitud

- Expresa el número 72_{10} en Signo-Magnitud con N=8:
  - Es positivo -> primer bit “0”
  - $|72_{10}| = 72_{10}$ con 7 bits ≡ $100\ 1000_2$

Signo-Magnitud

- Expresa el número 72_{10} en Signo-Magnitud con N=8:
  - Es positivo -> primer bit “0”
  - $| 72_{10} | = 72_{10}$ con 7 bits ≡ $100\ 1000_2$
- Resultado:
  - $72_{10} \equiv 0100\ 1000_2$

Signo-Magnitud

- Expresa el número -72_{10} en Signo-Magnitud con N=8:

Signo-Magnitud

- Expresa el número -72_{10} en Signo-Magnitud con N=8:
  - Es negativo -> primer bit “1”
  - $|-72_{10}| = 72_{10}$ con 7 bits ≡ $100\ 1000_2$

Signo-Magnitud

- Expresa el número -72_{10} en Signo-Magnitud con N=8:
  - Es negativo -> primer bit “1”
  - $|-72_{10}| = 72_{10}$ con 7 bits ≡ $100\ 1000_2$
- Resultado:
  - $-72_{10} \equiv 1100\ 1000_2$

Signo-Magnitud

- Expresa el número -111_{10} en Signo-Magnitud con N=8:

Signo-Magnitud

- Expresa el número -111_{10} en Signo-Magnitud con N=8:
  - Es negativo -> primer bit “1”
  - $|-111_{10}| = 111_{10}$ con 7 bits ≡ $110\ 1111_2$

Signo-Magnitud

- Expresa el número -111_{10} en Signo-Magnitud con N=8:
  - Es negativo -> primer bit “1”
  - $|-111_{10}| = 111_{10}$ con 7 bits -> $110\ 1111_2$
- Resultado:
  - $-111_{10} \equiv 1110\ 1111_2$

Signo-Magnitud

- Expresa el número 145_{10} en Signo-Magnitud con N=8:
  - Este número **no se puede expresar** en Signo-Magnitud con 8 bits

Signo-Magnitud

- El número $1110\ 0100_2$
- ¿es 228_{10} o -100_{10}?

Signo-Magnitud

- El número $1110\ 0100_2$
- ¿es 228_{10} o -100_{10}?
- Operación **con signo** o **sin signo**.

Signo-Magnitud

- ¿Cuál es el resultado de sumar los siguientes números binarios?
  - $0010\ 1111_2$
  - $1000\ 1100_2$

Signo-Magnitud

- ¿Cuál es el resultado de sumar los siguientes números binarios?

SIN SIGNO

- $0010\ 1111_2 \equiv 47_{10}$
- $1000\ 1100_2 \equiv 140_{10}$

Signo-Magnitud

- ¿Cuál es el resultado de sumar los siguientes números binarios?

  SIN SIGNO

  - $0010\ 1111_2 \equiv 47_{10}$
  - $1000\ 1100_2 \equiv 140_{10}$

- El resultado será:
  - $47_{10} + 140_{10} = 187_{10} \equiv 1011\ 1011_2$

Signo-Magnitud

- ¿Cuál es el resultado de sumar los siguientes números binarios?

CON SIGNO

- $0010\ 1111_2 \equiv 47_{10}$
- $1000\ 1100_2 \equiv -12_{10}$

Signo-Magnitud

- ¿Cuál es el resultado de sumar los siguientes números binarios?

  CON SIGNO

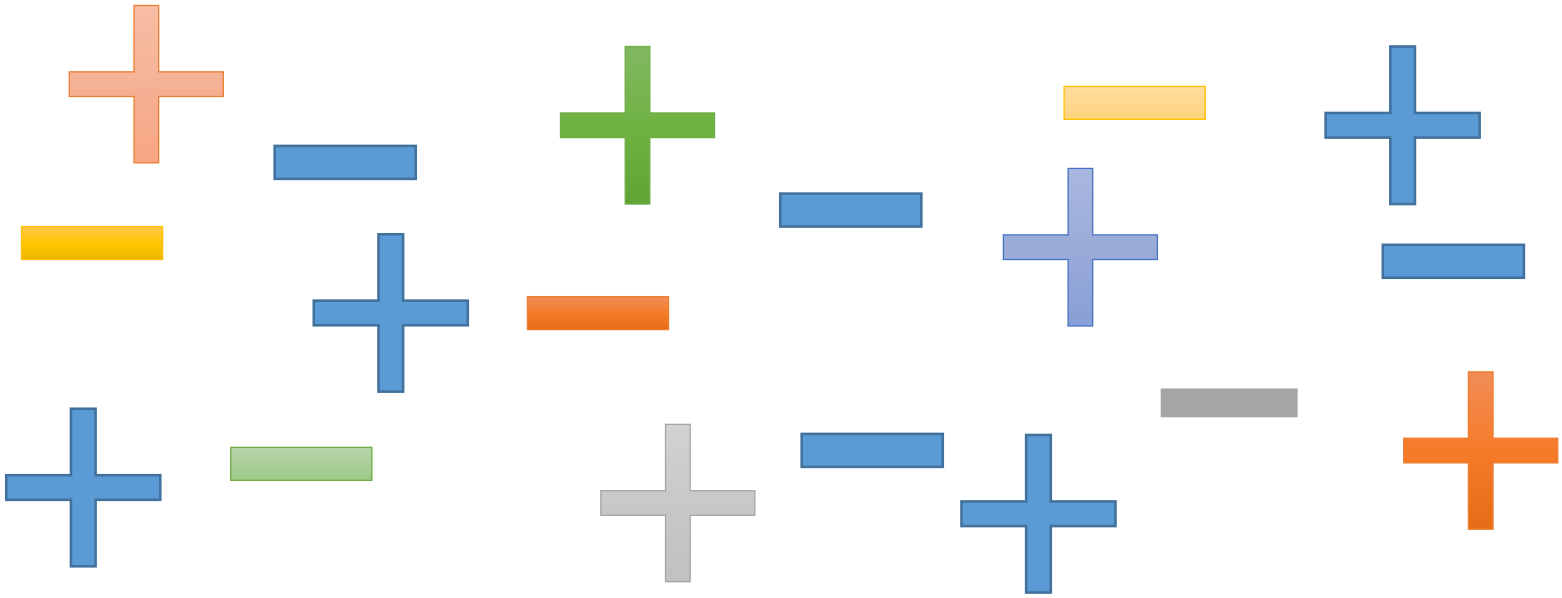
  - $0010\ 1111_2 \equiv 47_{10}$
  - $1000\ 1100_2 \equiv -12_{10}$

- El resultado será:
  - $47_{10} + -12_{10} = 35_{10} \equiv 0010\ 0011_2$

Signo-Magnitud

- Pregunta:
  - **¿Por qué no se usa?**
  - Doble representación del cero:
    - Con N=4, $0_{10} \equiv 0000_2$ y $0_{10} \equiv 1000_2$
  - Complicaciones en el diseño de los procesadores.

Codificación de números enteros con signo



Exceso-Z

Exceso Z

- Dado un número **X**, la representación en **Exceso Z** consiste en representar **X + Z** en binario.

- $Z = 2^{N-1}-1$

Exceso Z

- N = 4, Z = 7
- N = 8, Z = 127
- N = 16, Z = 32 767
- N = 32, Z = 2 147 483 647

Exceso Z

- Expresa el número -3_{10} en Exceso Z con N=4:

Exceso Z

- Expresa el número -3_{10} en Exceso Z con N=4:
  - Z = 7
  - $-3_{10} + 7_{10} = 4_{10}$

Exceso Z

- Expresa el número -3_{10} en Exceso Z con N=4:
  - $Z = 7$
  - $-3_{10} + 7_{10} = 4_{10}$
  - $4_{10} \equiv 0100_2$

Exceso Z

- Expresa el número -3_{10} en Exceso Z con N=4:
  - Z = 7
  - $-3_{10} + 7_{10} = 4_{10}$
  - $4_{10} \equiv 0100_2$
  - El resultado es: $-3_{10} \equiv 0100_2$

Exceso Z

- Expresa el número 5_{10} en Exceso Z con N=4:

Exceso Z

- Expresa el número 5_{10} en Exceso Z con N=4:
  - Z = 7
  - $5_{10} + 7_{10} = 12_{10}$

Exceso Z

- Expresa el número 5_{10} en Exceso Z con N=4:
  - $Z = 7$
  - $5_{10} + 7_{10} = 12_{10}$
  - $12_{10} \equiv 1100_2$

Exceso Z

- Expresa el número 5_{10} en Exceso Z con N=4:
  - Z = 7
  - $5_{10} + 7_{10} = 12_{10}$
  - $12_{10} \equiv 1100_2$
  - El resultado es: $5_{10} \equiv 1100_2$

Exceso Z

- Rango:

$$[\ -Z,\ +2^{N} - 1 - Z]$$

Exceso Z

- Con N=4, **sin signo**

$[0, +2^N-1]$ $[0, 15]$

- Con N=4, **con signo**, en Exceso Z (Z=7)

$[-Z, +2^N - 1 - Z]$ $[-7, +8]$

Exceso Z

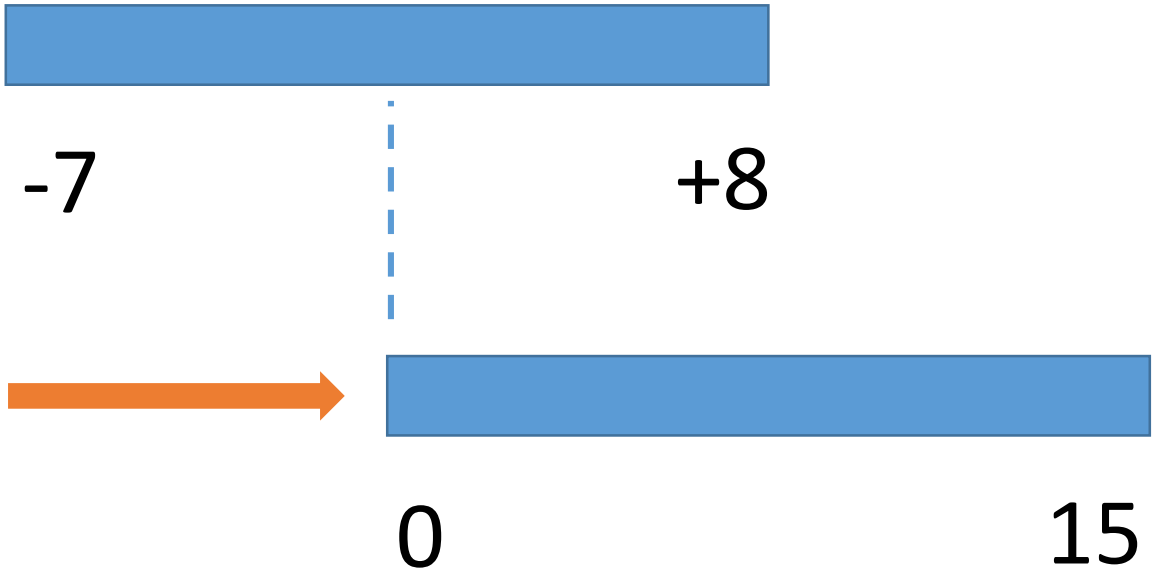
- Con N=8, **sin signo**

$[0, +2^N-1]$ $[0, 255]$

- Con N=8, **con signo**, en Exceso Z (Z=127)

$[-Z, +2^N - 1 - Z]$ $[-127, +128]$

Exceso Z



Exceso Z

- Pregunta: **¿Por qué no se usa?**
  - **Positivo:**
    - Los positivos empiezan por “1” y los negativos por “0”.
    - Única representación del cero.
  - **Negativo:**
    - Complicaciones en el diseño de los procesadores.

Exceso Z

- Se usa en el formato IEEE754 para codificar el exponente.

Exceso-Z
De decimal a binario

Exceso Z

- Expresa el número 15_{10} en Exceso Z con N=8:

Exceso Z

- Expresa el número 15_{10} en Exceso Z con N=8:
  - Z = 127
  - $15_{10} + 127_{10} = 142_{10}$

Exceso Z

- Expresa el número 15_{10} en Exceso Z con N=8:
  - Z = 127
  - $15_{10} + 127_{10} = 142_{10}$
  - $142_{10} \equiv 1000\ 1110_2$

Exceso Z

- Expresa el número 15_{10} en Exceso Z con N=8:
  - Z = 127
  - $15_{10} + 127_{10} = 142_{10}$
  - $142_{10} \equiv 1000\ 1110_2$
  - El resultado es: $15_{10} \equiv 1000\ 1110_2$

Exceso Z

- Expresa el número -115_{10} en Exceso Z con N=8:

Exceso Z

- Expresa el número -115_{10} en Exceso Z con N=8:
  - Z = 127
  - $-115_{10} + 127_{10} = 12_{10}$

Exceso Z

- Expresa el número -115_{10} en Exceso Z con N=8:
  - Z = 127
  - $-115_{10} + 127_{10} = 12_{10}$
  - $12_{10} \equiv 0000\ 1100_2$

Exceso Z

- Expresa el número -115_{10} en Exceso Z con N=8:
  - Z = 127
  - $-115_{10} + 127_{10} = 12_{10}$
  - $12_{10} \equiv 0000\ 1100_2$
  - El resultado es: $-115_{10} \equiv 0000\ 1100_2$

Exceso Z

- Expresa el número -107_{10} en Exceso Z con N=8:

Exceso Z

- Expresa el número -107_{10} en Exceso Z con N=8:
  - Z = 127
  - $-107_{10} + 127_{10} = 20_{10}$

Exceso Z

- Expresa el número -107_{10} en Exceso Z con N=8:
  - Z = 127
  - $-107_{10} + 127_{10} = 20_{10}$
  - $20_{10} \equiv 0001\ 0100_2$

Exceso Z

- Expresa el número -107_{10} en Exceso Z con N=8:
  - Z = 127
  - $-107_{10} + 127_{10} = 20_{10}$
  - $20_{10} \equiv 0001\ 0100_2$
  - El resultado es: $-107_{10} \equiv 0001\ 0100_2$

Exceso-Z
De binario a decimal

Exceso Z

- Convierte a decimal el número 1010_2 expresado en Exceso Z con N=4.

Exceso Z

- Convierte a decimal el número 1010_2 expresado en Exceso Z con N=4.
  - Z=7
  - $1010_2 \equiv 10_{10}$

Exceso Z

- Convierte a decimal el número 1010_2 expresado en Exceso Z con N=4.
  - Z=7
  - $1010_2 \equiv 10_{10}$
  - $10_{10} - 7_{10} = 3_{10}$

Exceso Z

- Convierte a decimal el número 1010_2 expresado en Exceso Z con N=4.
  - Z=7
  - $1010_2 \equiv 10_{10}$
  - $10_{10} - 7_{10} = 3_{10}$
  - El resultado es $1010_2 \equiv 3_{10}$

Exceso Z

- Convierte a decimal el número $0010\ 1010_2$ expresado en Exceso Z con N=8.

Exceso Z

- Convierte a decimal el número $0010\ 1010_2$ expresado en Exceso Z con N=8.
  - Z=127
  - $0010\ 1010_2 \equiv 42_{10}$

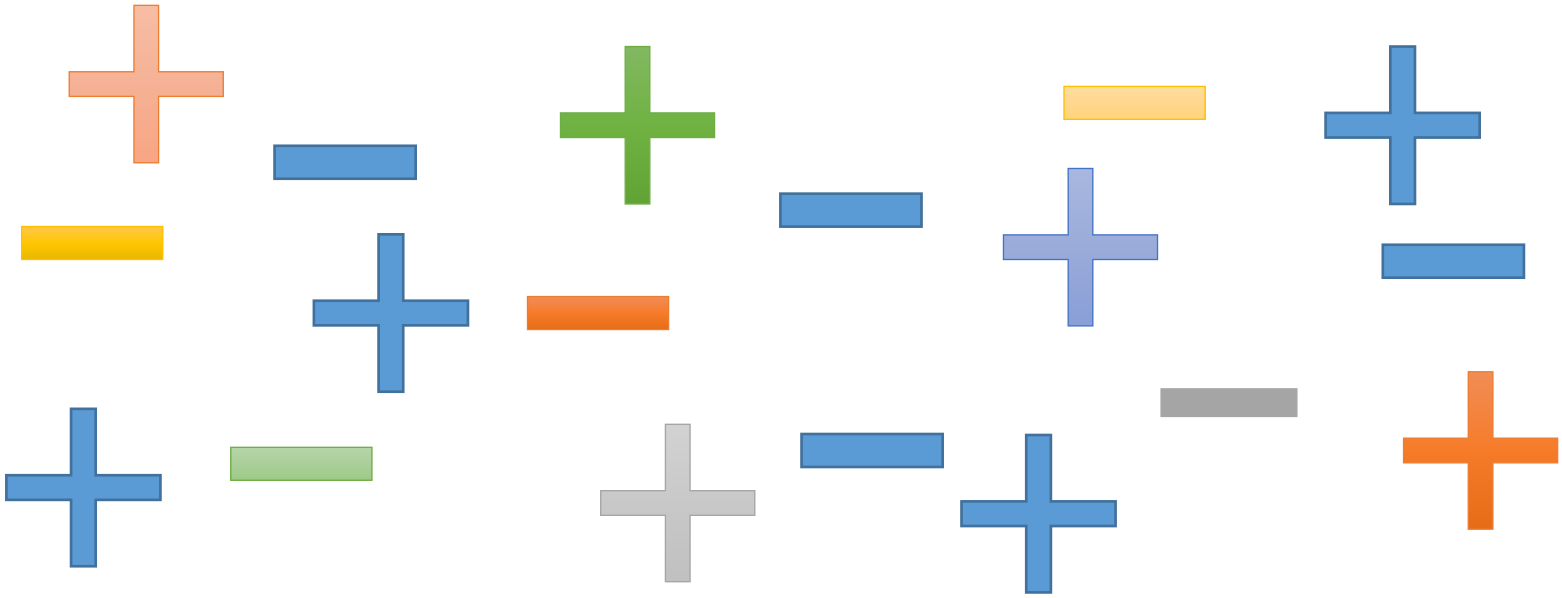
Exceso Z

- Convierte a decimal el número $0010\ 1010_2$ expresado en Exceso Z con N=8.
  - Z=127
  - $0010\ 1010_2 \equiv 42_{10}$
  - $42_{10} - 127_{10} = -85_{10}$

Exceso Z

- Convierte a decimal el número $0010\ 1010_2$ expresado en Exceso Z con N=8.
  - Z=127
  - $0010\ 1010_2 \equiv 42_{10}$
  - $42_{10} - 127_{10} = -85_{10}$
  - El resultado es $0010\ 1010_2 \equiv -85_{10}$

Codificación de números enteros con signo



Complemento a 2

Complemento a 2

- Dado un número **x**:
  - Si es **positivo**:
    - Binario
  - Si es **negativo**, tres pasos:
    1. Convertir |x| a binario
    2. Cambiar 0s por 1s y viceversa
    3. Sumar 1

Complemento a 2

- Expresa el número 3_{10} en Complemento a 2 con N=4:

Complemento a 2

- Expresa el número 3_{10} en Complemento a 2 con N=4:
  - Es positivo
  - $3_{10} \equiv 0011_2$

Complemento a 2

- Expresa el número 3_{10} en Complemento a 2 con N=4:
  - Es positivo
  - $3_{10} \equiv 0011_2$
- El resultado es: $3_{10} \equiv 0011_2$

Complemento a 2

- Expresa el número -3_{10} en Complemento a 2 con N=4:

Complemento a 2

- Expresa el número -3_{10} en Complemento a 2 con N=4:
  - Es negativo
    - Paso 1: $|-3_{10}| = 3_{10} \equiv 0011_2$
    - Paso 2: 0011_2 -> 1100_2
    - Paso 3: $1100_2 + 1_2 = 1101_2$
- El resultado es $-3_{10} \equiv 1101_2$

Complemento a 2

- Rango:

$$[-2^{N-1} , +2^{N-1} - 1]$$

Complemento a 2

- Con N=4, **sin signo**

$[0, +2^N-1]$ $[0, 15]$

- Con N=4, **con signo**, en Ca2

$[-2^{N-1} , +2^{N-1} - 1]$ $[-8, +7]$

Complemento a 2

- Con N=8, **sin signo**

$[0, +2^N-1]$ $[0, 255]$

- Con N=8, **con signo**, en Ca2

$[-2^{N-1} , +2^{N-1} - 1]$ $[-128, +127]$

Complemento a 2

- Pregunta: **¿Por qué se usa?**
  - **Positivo:**
    - Los positivos empiezan por “0” y los negativos por “1”.
    - Única representación del cero.
    - **Facilita** el diseño de los procesadores.

Complemento a 2

- Restar A – B, es equivalente a sumar A y el Ca2 de B.

Complemento a 2

- $A=5_{10}$, $B=3_{10}$ ¿A-B? con N=4

Complemento a 2

- $A=5_{10}$, $B=3_{10}$ ¿A-B? con N=4

- $5_{10} \equiv 0101_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

Complemento a 2

- $A=5_{10}$, $B=3_{10}$ ¿A-B? con N=4

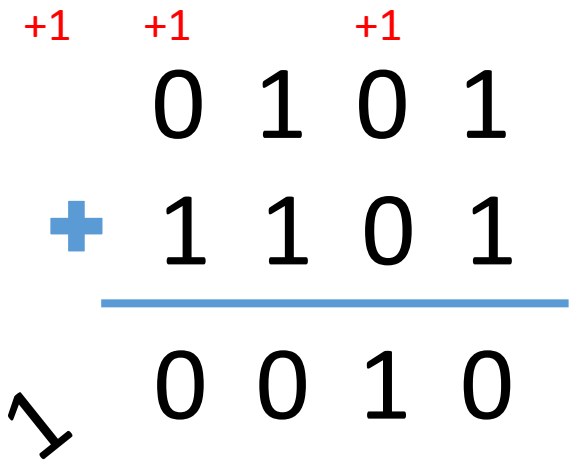
- $5_{10} \equiv 0101_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

$$\begin{array}{r} 0\ 1\ 0\ 1 \\ +\ 1\ 1\ 0\ 1 \\ \hline \end{array}$$

Complemento a 2

- $A=5_{10}$, $B=3_{10}$ ¿A-B? con N=4

- $5_{10} \equiv 0101_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

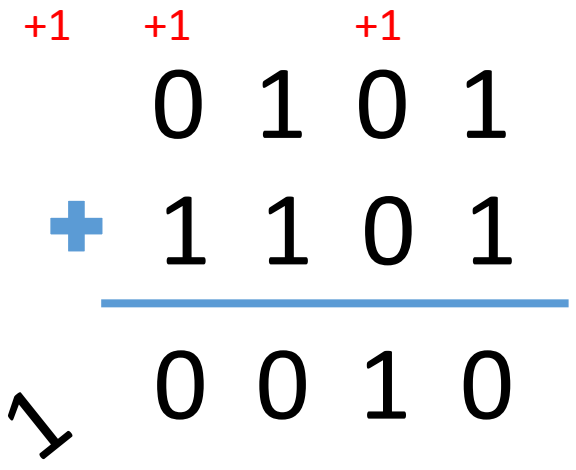


Complemento a 2

- A=5_{10}, B=3_{10} ¿A-B? con N=4

- $5_{10} \equiv 0101_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

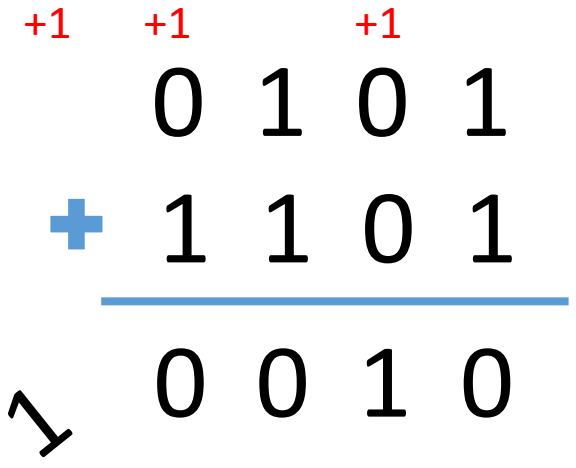
- $5_{10}-3_{10} = 2_{10}$
- $2_{10} \equiv 0010_2$



Complemento a 2

- $A=5_{10}$ con N=4
- $5_{10} \equiv 0101_2$
- $-3_{10} \equiv 1101_2$ (en C2)
- $5_{10}-3_{10} = 2_{10}$
- $2_{10} \equiv 0010_2$

Hay que descarta el último acarreo



Complemento a 2

- $A=2_{10}$, $B=3_{10}$ ¿A-B? con N=4

Complemento a 2

- $A=2_{10}$, $B=3_{10}$ ¿A-B? con N=4

- $2_{10} \equiv 0010_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

Complemento a 2

- $A=2_{10}$, $B=3_{10}$ ¿A-B? con N=4

- $2_{10} \equiv 0010_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

$$\begin{array}{r} 0\ 0\ 1\ 0 \\ +\ 1\ 1\ 0\ 1 \\ \hline \end{array}$$

Complemento a 2

- $A=2_{10}$, $B=3_{10}$ ¿A-B? con N=4

- $2_{10} \equiv 0010_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

$$\begin{array}{cccc} 0 & 0 & 1 & 0 \\ + \; 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 \end{array}$$

Complemento a 2

- $A=2_{10}$, $B=3_{10}$ ¿A-B? con N=4

- $2_{10} \equiv 0010_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

$$\begin{array}{cccc} 0 & 0 & 1 & 0 \\ + \; 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 \end{array}$$

- $2_{10}-3_{10} = -1_{10}$
- $-1_{10} \equiv 1111_2$ (en Ca2)

Complemento a 2

- $A=2_{10}$ con N=4
- $2_{10} \equiv 0010_2$
- $-3_{10} \equiv 1101_2$ (en Ca2)

No es necesario descarta el último acarreo

$$\begin{array}{cccc} 0 & 0 & 1 & 0 \\ + \; 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 \end{array}$$

- $2_{10}-3_{10} = -1_{10}$
- $-1_{10} \equiv 1111_2$ (en Ca2)

Complemento a 2

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

N=4

$3_{10} \equiv 0011_2$

N=8

$3_{10} \equiv 0000\ 0011_2$

Complemento a 2

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

N=4

$3_{10} \equiv 0011_2$

$5_{10} \equiv 0101_2$

N=8

$3_{10} \equiv 0000\ 0011_2$

$5_{10} \equiv 0000\ 0101_2$

Complemento a 2

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

N=4

$3_{10} \equiv 0011_2$

$5_{10} \equiv 0101_2$

$-2_{10} \equiv 1110_2$

N=8

$3_{10} \equiv 0000\ 0011_2$

$5_{10} \equiv 0000\ 0101_2$

$-2_{10} \equiv 1111\ 1110_2$

Complemento a 2

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

N=4

$3_{10} \equiv 0011_2$

$5_{10} \equiv 0101_2$

$-2_{10} \equiv 1110_2$

$-6_{10} \equiv 1010_2$

N=8

$3_{10} \equiv 0000\ 0011_2$

$5_{10} \equiv 0000\ 0101_2$

$-2_{10} \equiv 1111\ 1110_2$

$-6_{10} \equiv 1111\ 1010_2$

Complemento a 2
De decimal a binario

Complemento a 2

- Expresa el número 15_{10} en Ca2 con N=8

Complemento a 2

- Expresa el número 15_{10} en Ca2 con N=8
  - Es positivo
  - $15_{10} \equiv 0000\ 1111_2$

Complemento a 2

- Expresa el número 115_{10} en Ca2 con N=8

Complemento a 2

- Expresa el número 115_{10} en Ca2 con N=8
  - Es positivo
  - $115_{10} \equiv 0111\ 0011_2$

Complemento a 2

- Expresa el número -15_{10} en Ca2 con N=8

Complemento a 2

- Expresa el número -15_{10} en Ca2 con N=8
  - Es negativo
    - Paso 1: $|-15_{10}| = 15_{10} \equiv 0000\ 1111_2$
    - Paso 2: $0000\ 1111_2$ -> $1111\ 0000_2$
    - Paso 3: $1111\ 0000_2 + 1_2 = 1111\ 0001_2$

Complemento a 2

- Expresa el número -15_{10} en Ca2 con N=8
  - Es negativo
    - Paso 1: $|-15_{10}| = 15_{10} \equiv 0000\ 1111_2$
    - Paso 2: $0000\ 1111_2$ -> $1111\ 0000_2$
    - Paso 3: $1111\ 0000_2 + 1_2 = 1111\ 0001_2$
- El resultado es $-15_{10} \equiv 1111\ 0001_2$

Complemento a 2

- Expresa el número -105_{10} en Ca2 con N=8

Complemento a 2

- Expresa el número -105_{10} en Ca2 con N=8
  - Es negativo
    - Paso 1: $|-105_{10}| = 105_{10} \equiv 0110\ 1001_2$
    - Paso 2: $0110\ 1001_2$ -> $1001\ 0110_2$
    - Paso 3: $1001\ 0110_2 + 1_2 = 1001\ 0111_2$

Complemento a 2

- Expresa el número -105_{10} en Ca2 con N=8
  - Es negativo
    - Paso 1: $|-105_{10}| = 105_{10} \equiv 0110\ 1001_2$
    - Paso 2: $0110\ 1001_2$ -> $1001\ 0110_2$
    - Paso 3: $1001\ 0110_2 + 1_2 = 1001\ 0111_2$
- El resultado es $-105_{10} \equiv 1001\ 0111_2$

Complemento a 2
De binario a decimal

Complemento a 2

- Convierte a decimal el número 0010_2 expresado en Ca2 con N=4.

Complemento a 2

- Convierte a decimal el número 0010_2 expresado en Ca2 con N=4.
  - Como empieza por 0 es positivo
  - $0010_2 \equiv 2_{10}$

Complemento a 2

- Convierte a decimal el número 1011_2 expresado en Ca2 con N=4.

Complemento a 2

- Convierte a decimal el número 1011_2 expresado en Ca2 con N=4.
  - Como empieza por 1 es negativo (tres pasos al revés):
    - Paso 1: $1011_2 - 1_2 = 1010_2$
    - Paso 2: $1010_2 \rightarrow 0101_2$
    - Paso 3: $0101_2 \equiv 5_{10}$

Complemento a 2

- Convierte a decimal el número 1011_2 expresado en Ca2 con N=4.
  - Como empieza por 1 es negativo (tres pasos al revés):
    - Paso 1: $1011_2 - 1_2 = 1010_2$
    - Paso 2: $1010_2 \rightarrow 0101_2$
    - Paso 3: $0101_2 \equiv 5_{10}$
  - El resultado es $1011_2 \equiv 5_{10}$

Complemento a 2

- Convierte a decimal el número 1010 0010_2 expresado en Ca2 con N=8.

Complemento a 2

- Convierte a decimal el número $1010\ 0010_2$ expresado en Ca2 con N=8.
  - Como empieza por 1 es negativo (tres pasos al revés):
    - Paso 1: $1010\ 0010_2 - 1_2 = 1010\ 0001_2$
    - Paso 2: $1010\ 0001_2 \rightarrow 0101\ 1110_2$
    - Paso 3: $0101\ 1110_2 \equiv 94_{10}$

Complemento a 2

- Convierte a decimal el número $1010\ 0010_2$ expresado en Ca2 con N=8.
  - Como empieza por 1 es negativo (tres pasos al revés):
    - Paso 1: $1010\ 0010_2 - 1_2 = 1010\ 0001_2$
    - Paso 2: $1010\ 0001_2 \rightarrow 0101\ 1110_2$
    - Paso 3: $0101\ 1110_2 \equiv 94_{10}$
  - El resultado es $1010\ 0010_2 \equiv 94_{10}$

Codificación de números reales

Números reales

- 4,5
  - Parte entera: 4
  - Parte fraccionaria: 0,5

Números reales

- No se pueden expresar números reales en binario como: 100,1

- No es posible usar el símbolo “ , ”
  - Únicamente 0’s y 1’s

Números reales

- Estándar IEEE754
  - 32 bits
  - 64 bits

Notación científica
Decimal

Notación científica decimal

- Cualquier número se puede expresar como:
  - $a * b^e$
- Mantisa **a**: Número real con un único dígito en la parte entera.
- Exponente **e**
- Base **b**

Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica

5 8 1 , 8 7

Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica

5 8 1 , 8 7

/ 10

5 8 , 1 8 7

Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica

La coma se mueve hacia la izquierda

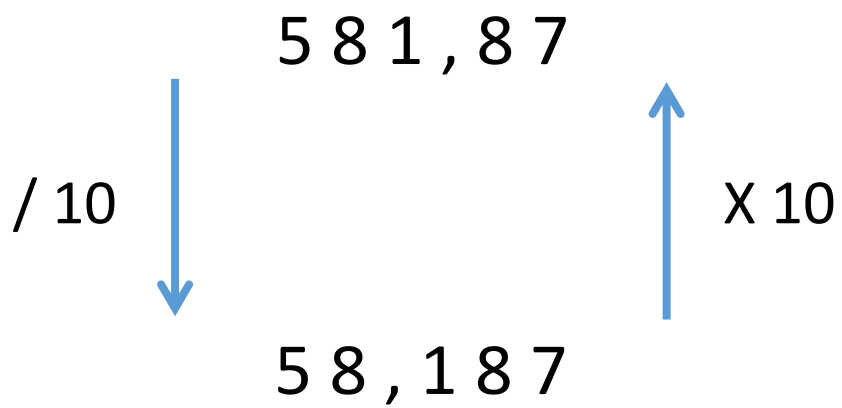
5 8 1 , 8 7

/ 10

5 8 , 1 8 7

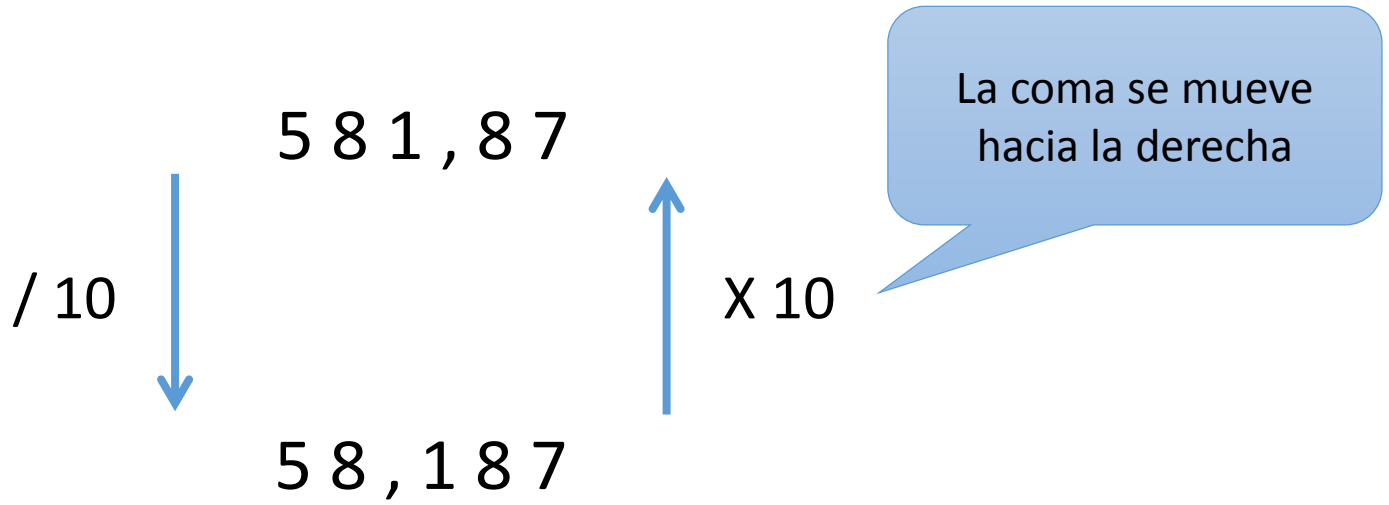
Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica



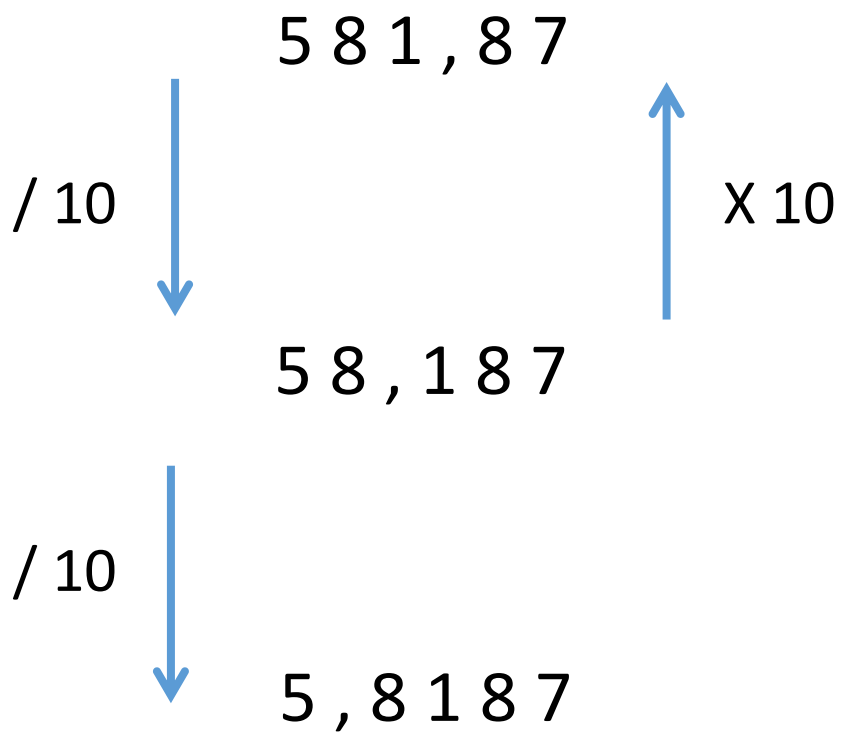
Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica



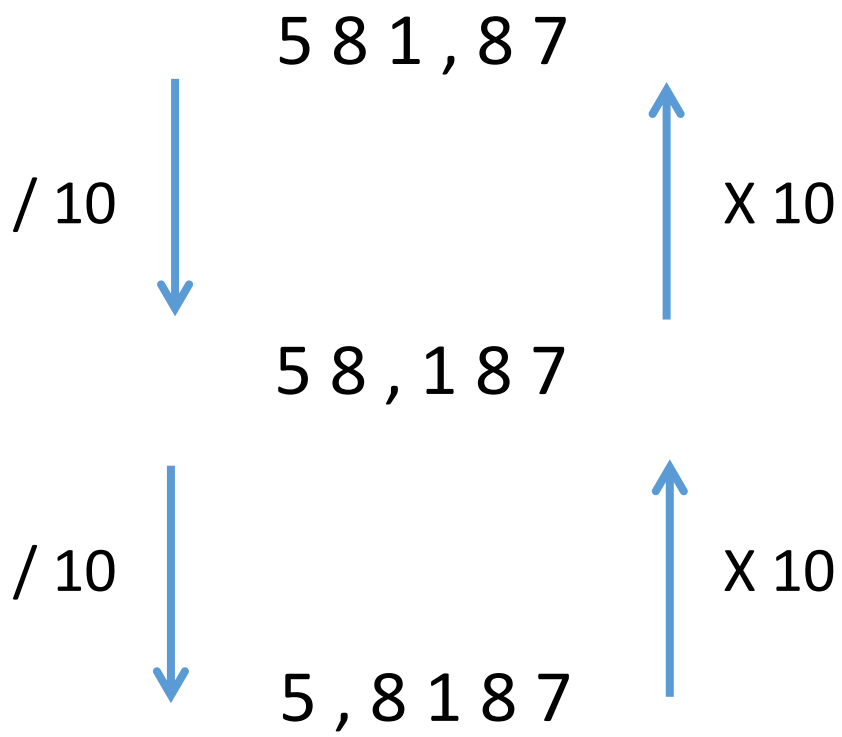
Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica



Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica



Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica

5 8 1 , 8 7

/ 10^2

5 , 8 1 8 7

Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica

La coma se mueve **2 veces** hacia la izquierda

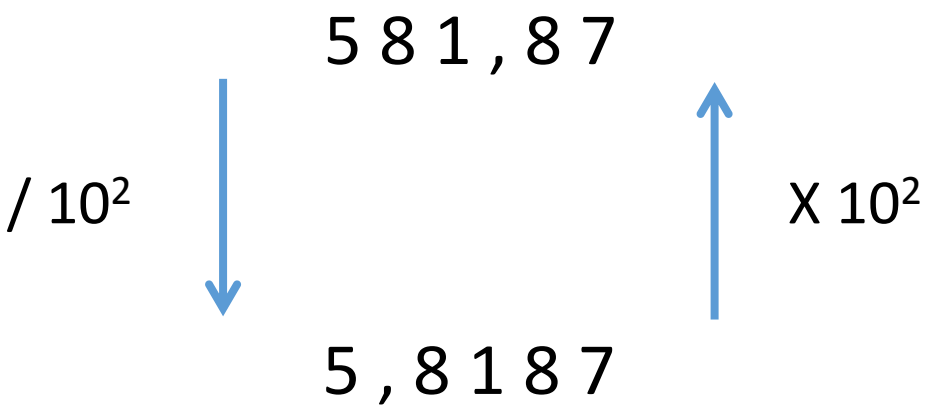
$5\ 8\ 1\ ,\ 8\ 7$

$/\ 10^2$

$5\ ,\ 8\ 1\ 8\ 7$

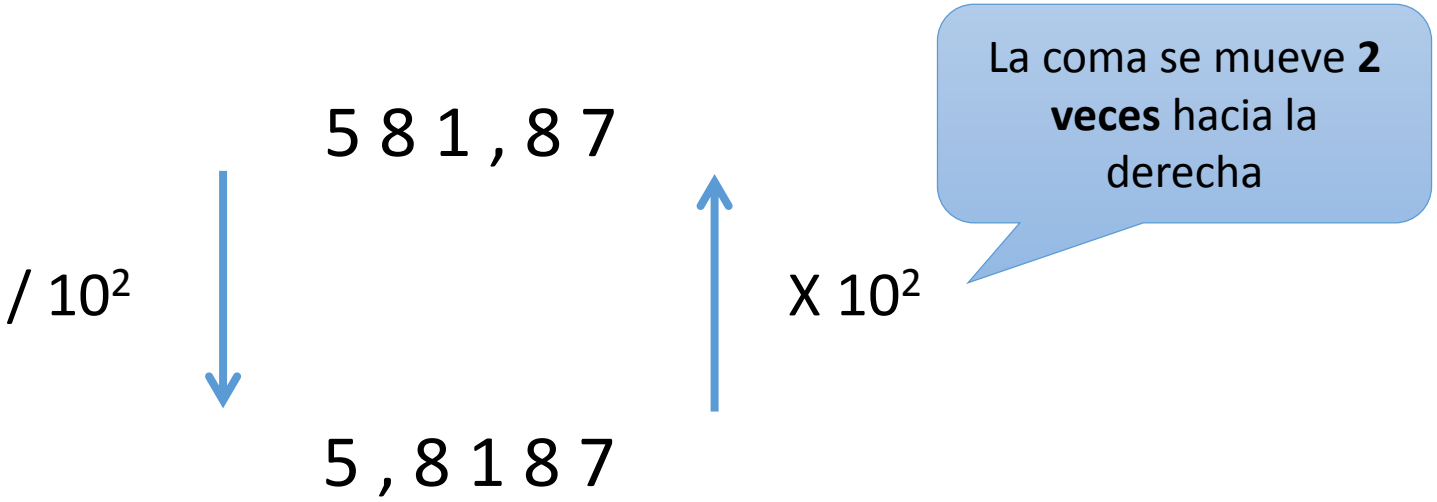
Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica



Notación científica decimal

- Expresa el número $581{,}87_{10}$ en notación científica



Notación científica decimal

- $581{,}87_{10} = 5{,}8187_{10} * 10^{2}$
  - Mantisa: $5{,}8187_{10}$
  - Base: 10_{10}
  - Exponente: 2_{10}

Notación científica decimal

- $1521{,}0687_{10} = 1{,}5210687_{10} * 10^{3}$
  - Mantisa: $1{,}5210687_{10}$
  - Base: 10_{10}
  - Exponente: 3_{10}

Notación científica decimal

- $0{,}00564_{10} = 5{,}64_{10} * 10^{-3}$
  - Mantisa: $5{,}64_{10}$
  - Base: 10_{10}
  - Exponente: -3_{10}

Notación científica decimal

- $0{,}00564_{10} = 5{,}64_{10} * 10^{-3}$
  - Mantisa: $5{,}64_{10}$
  - Base: 10_{10}
  - Exponente: -3_{10}

$5{,}64 * 10^{-3}$
=
$5{,}64 / 10^{3}$

Notación científica decimal

- $100000000000000_{10} = 1{,}0_{10} * 10^{14}$
  - Mantisa: $1{,}0_{10}$
  - Base: 10_{10}
  - Exponente: 14_{10}

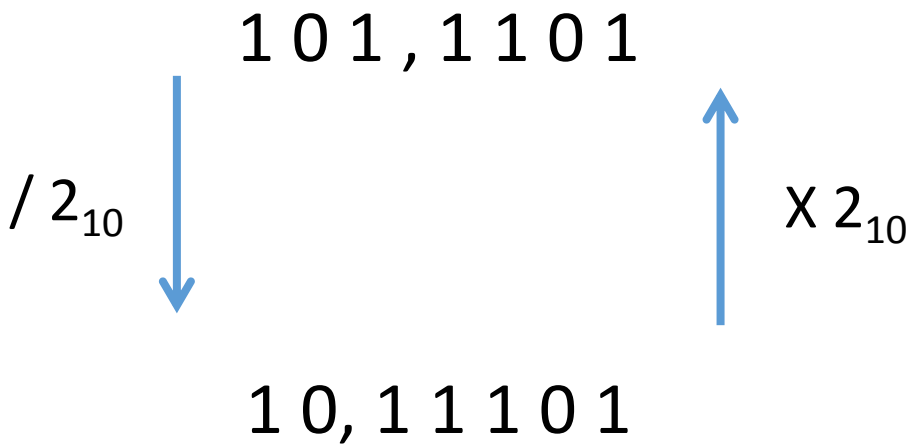
Notación científica decimal

- $0{,}00000000003_{10} = 3{,}0_{10} * 10^{-11}$
  - Mantisa: $3{,}0_{10}$
  - Base: 10_{10}
  - Exponente: -11_{10}

Notación científica Binario

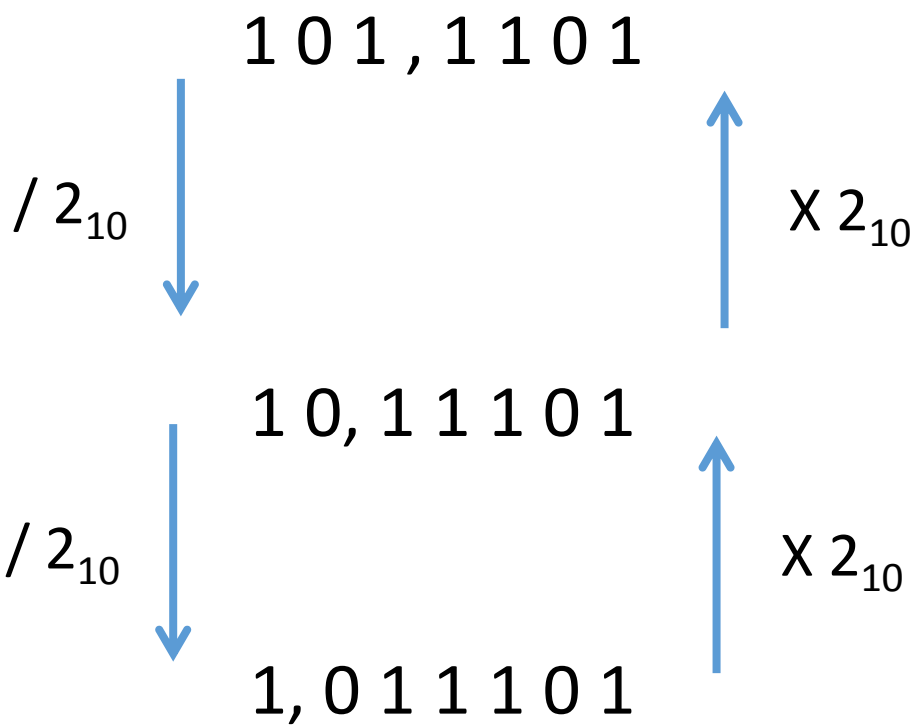
Notación científica binario

- Expresa el número $101{,}1101_2$ en notación científica



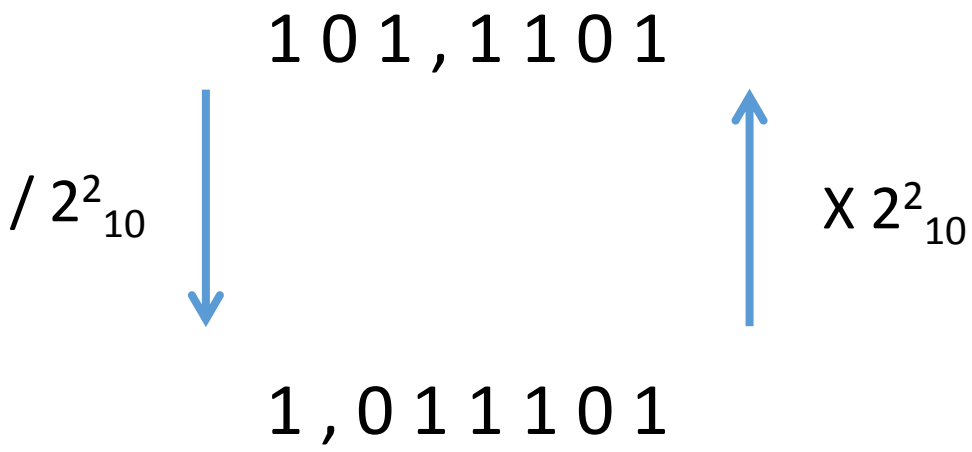
Notación científica binario

- Expresa el número $101{,}1101_2$ en notación científica



Notación científica binario

- Expresa el número $101{,}1101_2$ en notación científica



Notación científica binario

- $101{,}1101_2 \rightarrow 1{,}011101_2 * 2^2$
  - Mantisa: $1{,}011101_2$
  - Base: 2_{10}
  - Exponente: 2_{10}

Notación científica binario

ATENCIÓN

$$1{,}011101_2 * 2^2$$

Binario Decimal

Notación científica binario

ATENCIÓN

$$1{,}011101_2 * 2^2$$

Binario Decimal

$$101{,}1101_2 \rightarrow 1{,}011101_2 * 2^2$$

Notación científica binario

- $10111{,}10101_2 \rightarrow 1{,}011110101_2 * 2^4$
  - Mantisa: $1{,}011110101_2$
  - Base: 2_{10}
  - Exponente: 4_{10}

Notación científica binario

- $1000000000000_2 \rightarrow 1{,}0_2 * 2^{12}$
  - Mantisa: $1{,}0_2$
  - Base: 2_{10}
  - Exponente: 12_{10}

Notación científica binario

- $0{,}0011011_2 \rightarrow 1{,}1011_2 * 2^{-3}$
  - Mantisa: $1{,}1011_2$
  - Base: 2_{10}
  - Exponente: -3_{10}

Codificación de números reales

Preparando formato IEEE754

Formato IEEE754

- Pasos para codificar un número real en binario usando IEEE754:

1. Codificar en binario la parte entera
2. Codificar en binario la parte fraccionaria
3. Representar en notación científica binaria el número resultante de la suma de los pasos 1 y 2.
4. Aplicar el formato IEEE754

Formato IEEE754

- Expresa el número $5{,}375_{10}$ en binario usando notación científica:

Formato IEEE754

- Expresa el número $5{,}375_{10}$ en binario usando notación científica:
  - $5_{10} \equiv 101_2$
  - $0{,}375_{10} \equiv 0{,}011_2$

Formato IEEE754

- Expresa el número $5{,}375_{10}$ en binario usando notación científica:
  - $5_{10} \equiv 101_2$
  - $0{,}375_{10} \equiv 0{,}011_2$

$0{,}01_2 \equiv 0{,}25_{10}$
$0{,}001_2 \equiv 0{,}125_{10}$

Formato IEEE754

- Expresa el número $5{,}375_{10}$ en binario usando notación científica:
  - $5_{10} \equiv 101_2$
  - $0{,}375_{10} \equiv 0{,}011_2$
- $5{,}375_{10} \equiv 101{,}011_2$

Formato IEEE754

- Expresa el número $5{,}375_{10}$ en binario usando notación científica:
  - $5_{10} \equiv 101_2$
  - $0{,}375_{10} \equiv 0{,}011_2$
- $5{,}375_{10} \equiv 101{,}011_2$

- $101{,}011_2 \rightarrow 1{,}01011_2 * 2^2$

Resultado: $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$

Formato IEEE754

- Expresa el número $9{,}625_{10}$ en binario usando notación científica:

Formato IEEE754

- Expresa el número $9{,}625_{10}$ en binario usando notación científica:
  - $9_{10} \equiv 1001_2$
  - $0{,}625_{10} \equiv 0{,}101_2$

Formato IEEE754

- Expresa el número $9{,}625_{10}$ en binario usando notación científica:
  - $9_{10} \equiv 1001_2$
  - $0{,}625_{10} \equiv 0{,}101_2$
- $9{,}625_{10} \equiv 1001{,}101_2$

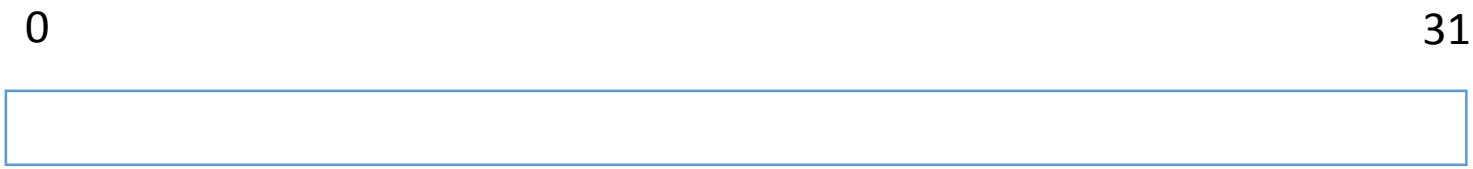
Formato IEEE754

- Expresa el número $9{,}625_{10}$ en binario usando notación científica:
  - $9_{10} \equiv 1001_2$
  - $0{,}625_{10} \equiv 0{,}101_2$
- $9{,}625_{10} \equiv 1001{,}101_2$
- $1001{,}101_2 \rightarrow 1{,}001101_2 * 2^3$

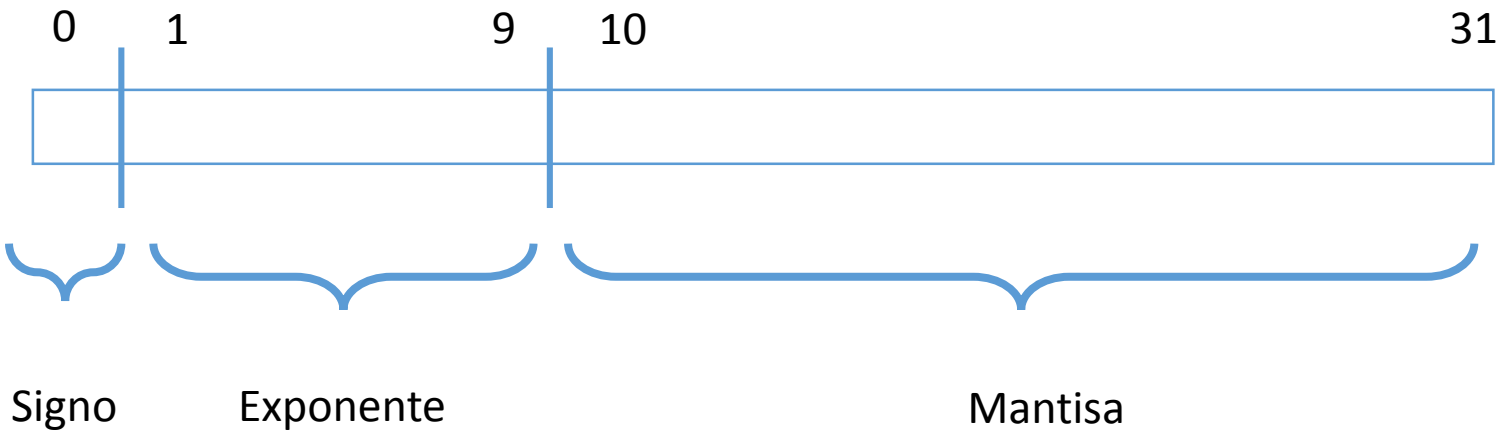
Resultado: $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$

IEEE754 32 bits

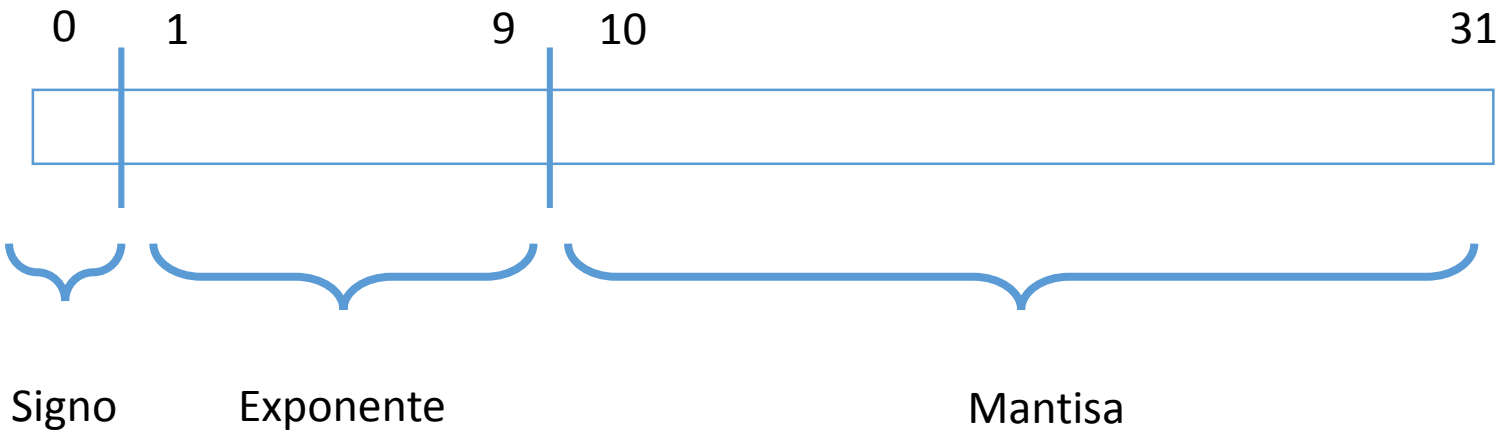
Formato IEEE754 32 bits



Formato IEEE754 32 bits



Formato IEEE754 32 bits



- **Signo**: 1 bit, 0 positivo, 1 negativo
- **Exponente**: 8 bits, en Exceso Z, Z=127
- **Mantisa**: 23 bits. Únicamente dígitos a la derecha de la coma
  (Todos números binarios reales empiezan por 1,…)

Formato IEEE754 32 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 32 bits:

Formato IEEE754 32 bits

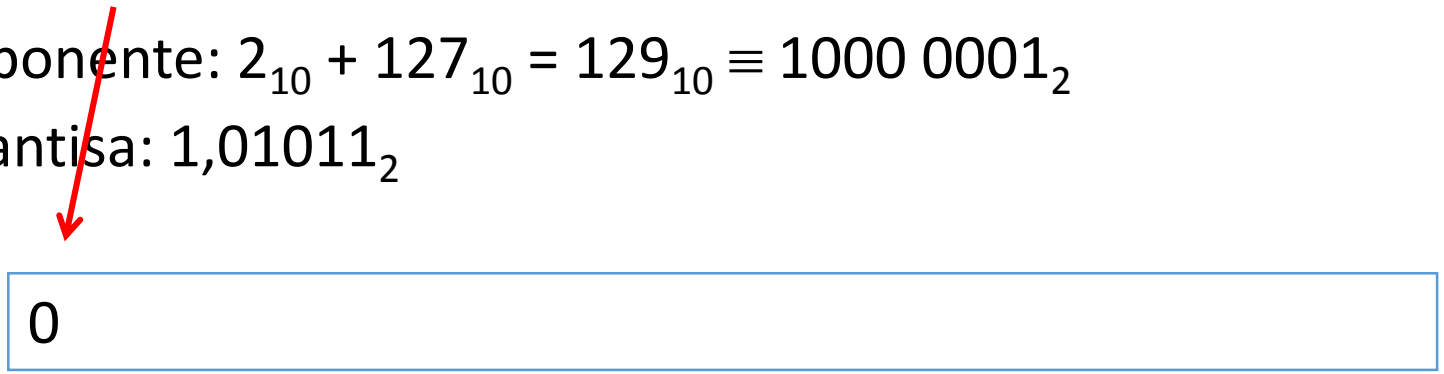
- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 32 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$

Formato IEEE754 32 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 32 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 127_{10} = 129_{10} \equiv 1000\ 0001_2$
- Mantisa: $1{,}01011_2$

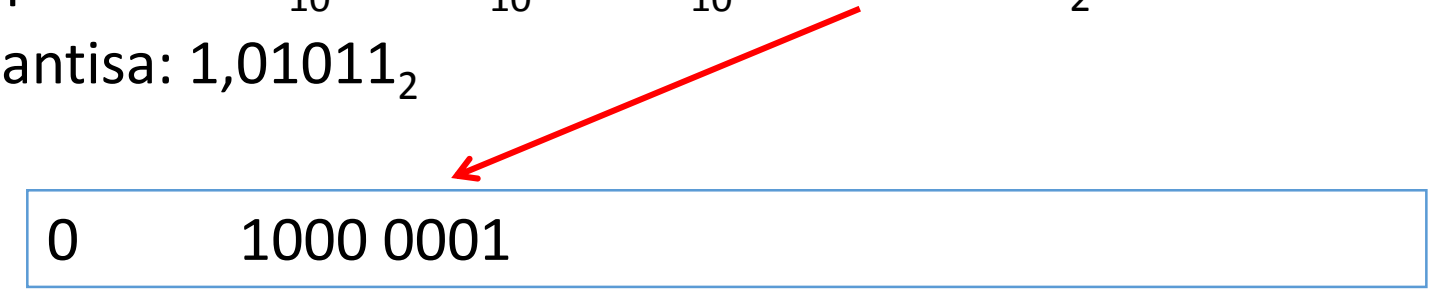
Formato IEEE754 32 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 32 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 127_{10} = 129_{10} \equiv 1000\ 0001_2$
- Mantisa: $1{,}01011_2$



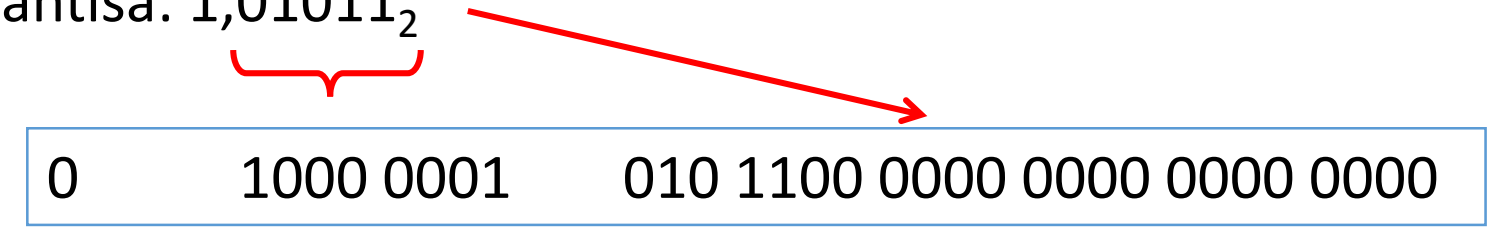
Formato IEEE754 32 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 32 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 127_{10} = 129_{10} \equiv 1000\ 0001_2$
- Mantisa: $1{,}01011_2$



Formato IEEE754 32 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 32 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 127_{10} = 129_{10} \equiv 1000\ 0001_2$
- Mantisa: $1{,}01011_2$



Formato IEEE754 32 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 32 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 127_{10} = 129_{10} \equiv 1000\ 0001_2$
- Mantisa: $1{,}01011_2$

0100 0000 1010 1100 0000 0000 0000 0000

40AC0000

Formato IEEE754 32 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 32 bits:

Solución: $5{,}375_{10} \equiv 40AC0000_{16}$

Formato IEEE754 32 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 32 bits:

Formato IEEE754 32 bits

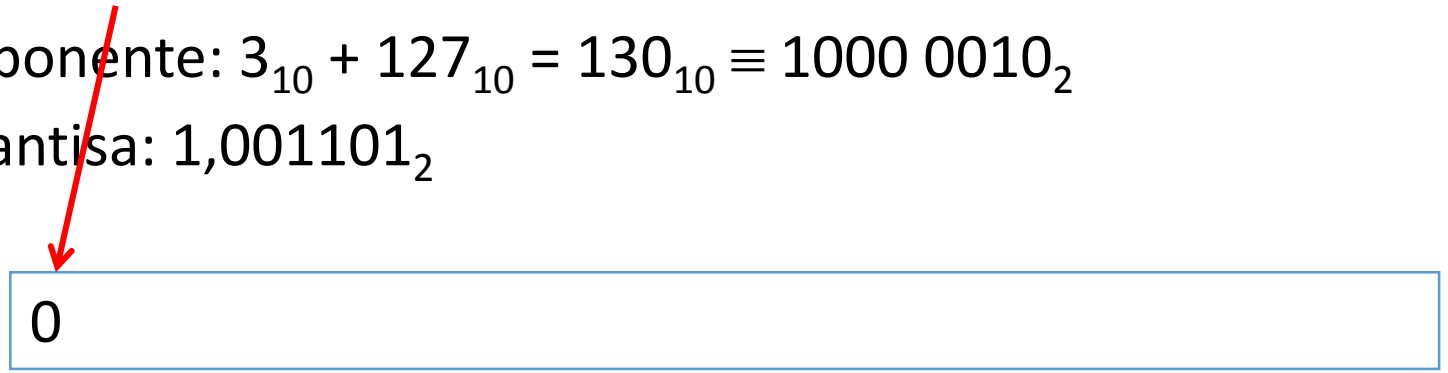
- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 32 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$

Formato IEEE754 32 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 32 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 127_{10} = 130_{10} \equiv 1000\ 0010_2$
- Mantisa: $1{,}001101_2$

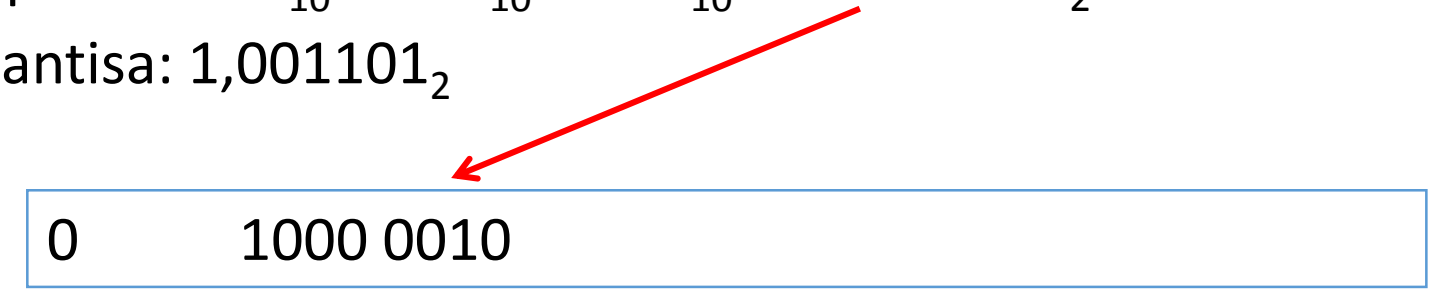
Formato IEEE754 32 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 32 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 127_{10} = 130_{10} \equiv 1000\ 0010_2$
- Mantisa: $1{,}001101_2$



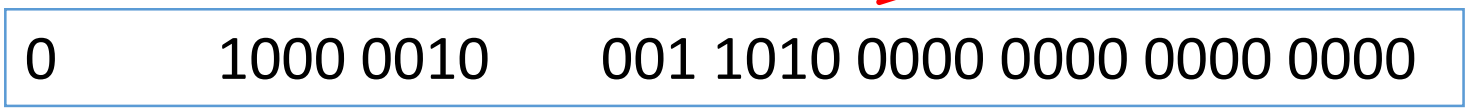
Formato IEEE754 32 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 32 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 127_{10} = 130_{10} \equiv 1000\ 0010_2$
- Mantisa: $1{,}001101_2$



Formato IEEE754 32 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 32 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 127_{10} = 130_{10} \equiv 1000\ 0010_2$
- Mantisa: $1{,}001101_2$



Formato IEEE754 32 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 32 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 127_{10} = 130_{10} \equiv 1000\ 0011_2$
- Mantisa: $1{,}001101_2$

0100 0001 0001 1010 0000 0000 0000 0000

411A0000

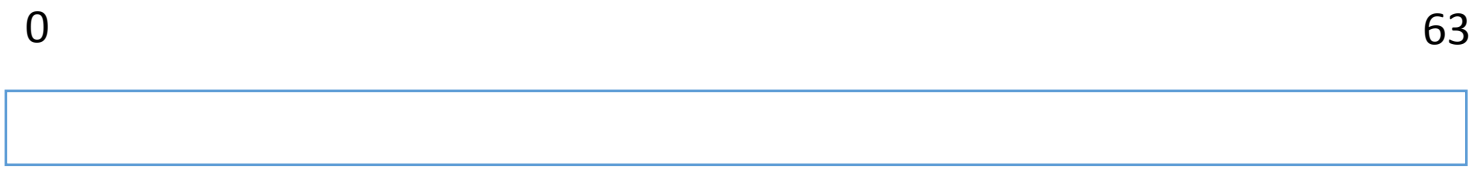
Formato IEEE754 32 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 32 bits:

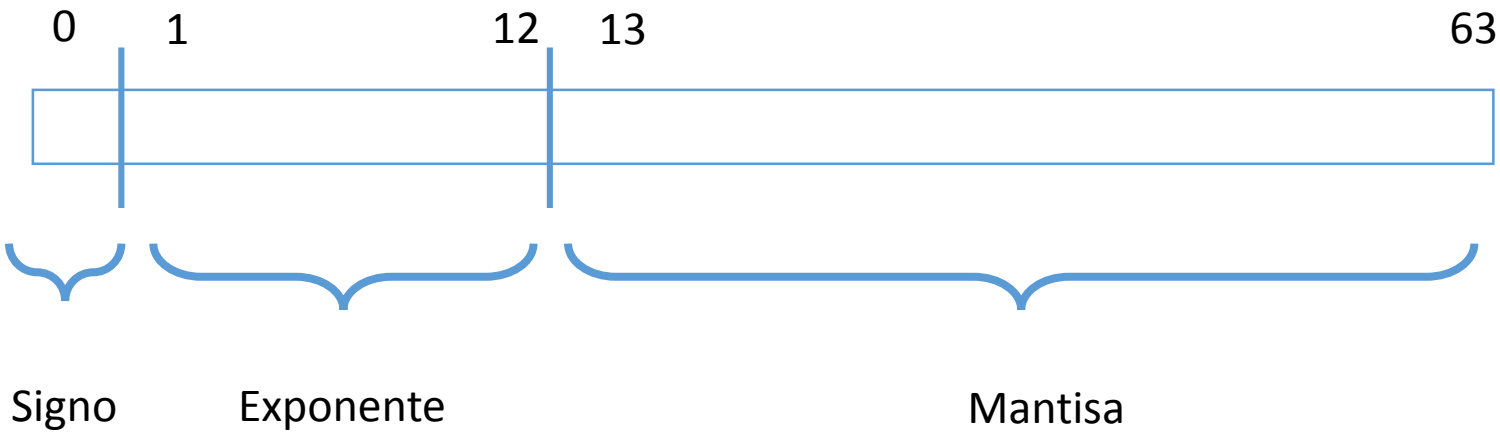
Solución: $9{,}625_{10} \equiv 409A0000_{16}$

IEEE754 64 bits

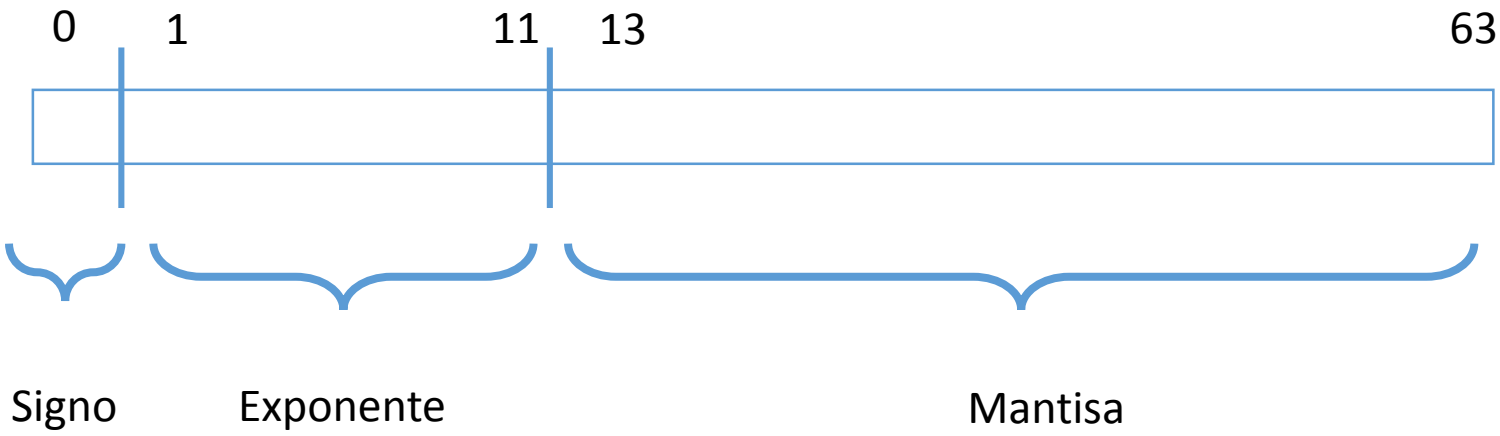
Formato IEEE754 64 bits



Formato IEEE754 64 bits

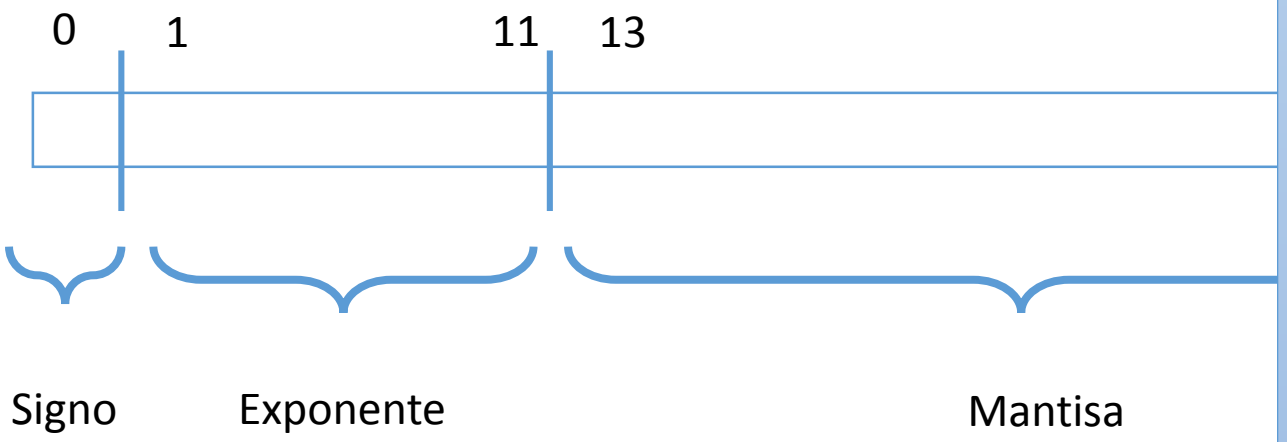


Formato IEEE754 64 bits



- **Signo**: 1 bit, 0 positivo, 1 negativo
- **Exponente**: 11 bits, en Exceso Z, Z=1023
- **Mantisa**: 52 bits. Únicamente dígitos a la derecha de la coma.

Formato IEEE754 64 bits



¿1023?

$2^{N-1}-1$

=

$2^{11-1}-1$

=

$2^{10}-1$

=

1024 - 1

- **Signo**: 1 bit, 0 positivo, 1 negativo
- **Exponente**: 11 bits, en Exceso Z, Z=1023
- **Mantisa**: 52 bits. Únicamente dígitos a la der
  la coma.

Formato IEEE754 64 bits

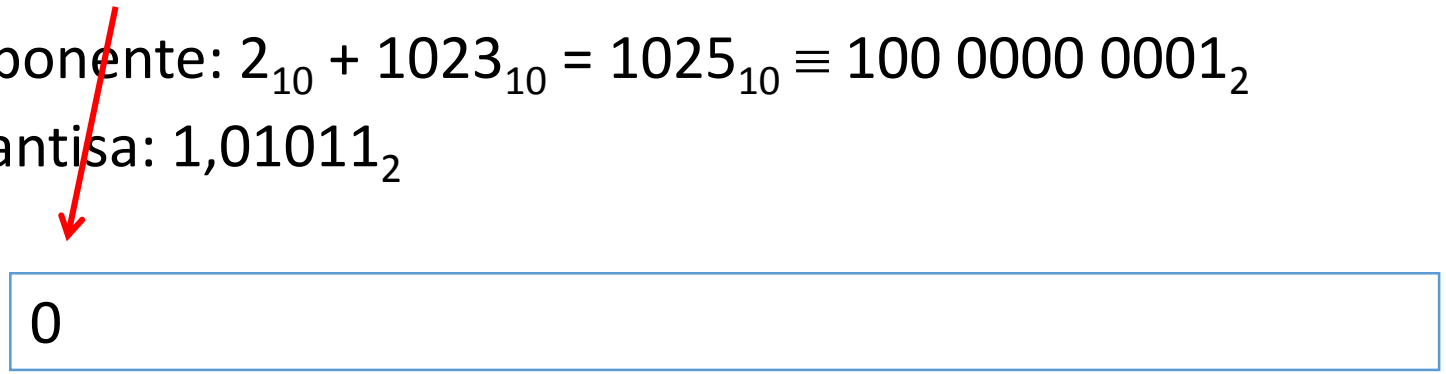
- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 64 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$

Formato IEEE754 64 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 64 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 1023_{10} = 1025_{10} \equiv 100\ 0000\ 0001_2$
- Mantisa: $1{,}01011_2$

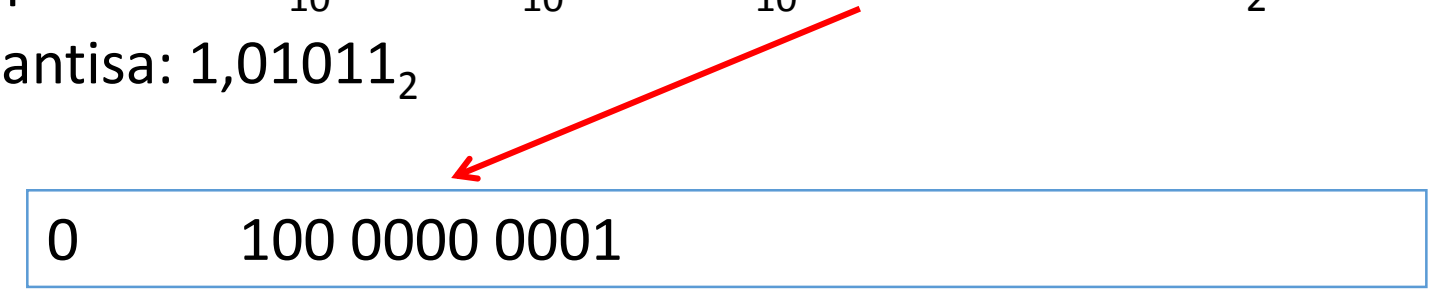
Formato IEEE754 64 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 64 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 1023_{10} = 1025_{10} \equiv 100\ 0000\ 0001_2$
- Mantisa: $1{,}01011_2$



Formato IEEE754 64 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 64 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 1023_{10} = 1025_{10} \equiv 100\ 0000\ 0001_2$
- Mantisa: $1{,}01011_2$



Formato IEEE754 64 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 64 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 1023_{10} = 1025_{10} \equiv 100\ 0000\ 0001_2$
- Mantisa: $1{,}01011_2$

0 100 0000 0001 0101 1000 0000 0000 0000 …

Formato IEEE754 64 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 64 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 1023_{10} = 1025$
- Mantisa: $1{,}01011_2$

¿…?

0's hasta completar los 52 dígitos

0 100 0000 0001 0101 1000 0000 0000 0000 …

Formato IEEE754 64 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 64 bits:
  - $5{,}375_{10} \rightarrow 1{,}01011_2 * 2^2$
- Signo: 0
- Exponente: $2_{10} + 1023_{10} = 1025_{10} \equiv 100\ 0000\ 0001_2$
- Mantisa: $1{,}01011_2$

0100 0000 0001 0101 1000 0000 0000 0000 …

4015800000000000

Formato IEEE754 64 bits

- Expresa el número $5{,}375_{10}$ en binario usando IEEE754 de 64 bits:

Solución: $5{,}375_{10} \equiv 4015800000000000_{16}$

Formato IEEE754 64 bits

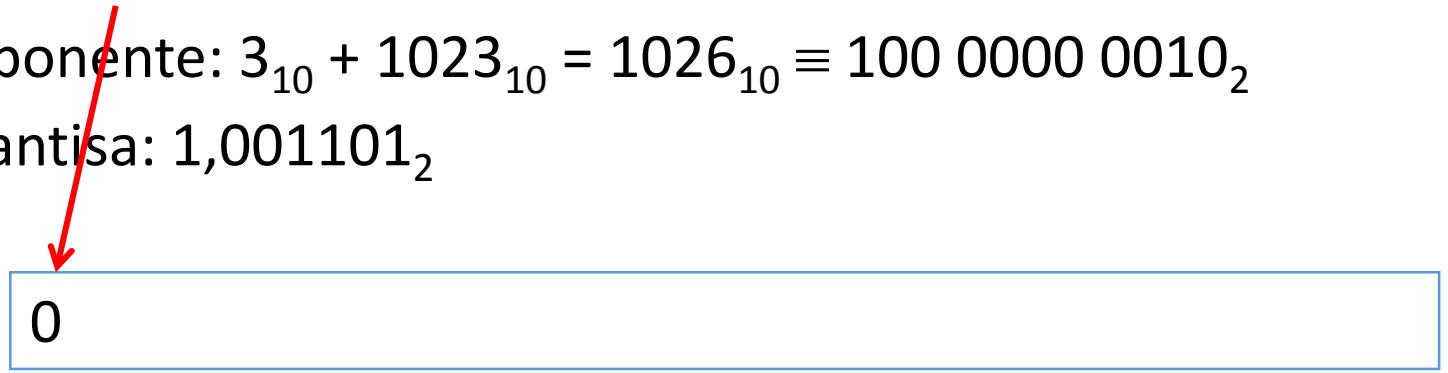
- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 64 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$

Formato IEEE754 64 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 64 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 1023_{10} = 1026_{10} \equiv 100\ 0000\ 0010_2$
- Mantisa: $1{,}001101_2$

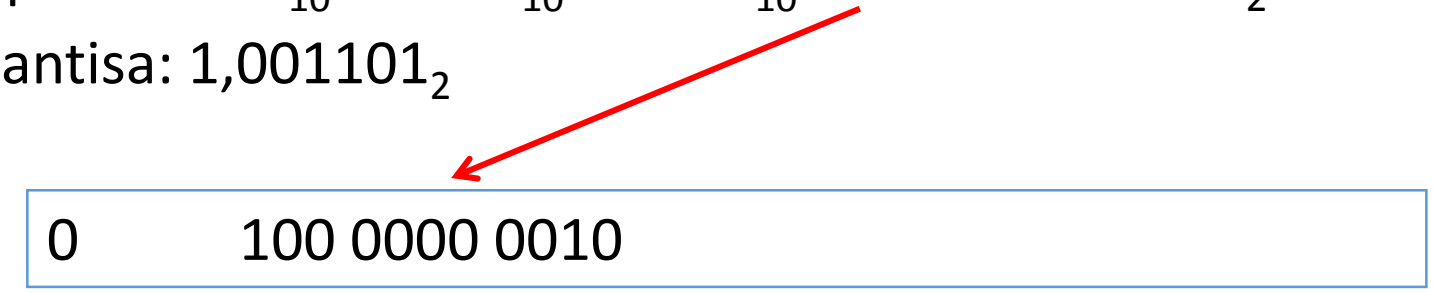
Formato IEEE754 64 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 64 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 1023_{10} = 1026_{10} \equiv 100\ 0000\ 0010_2$
- Mantisa: $1{,}001101_2$



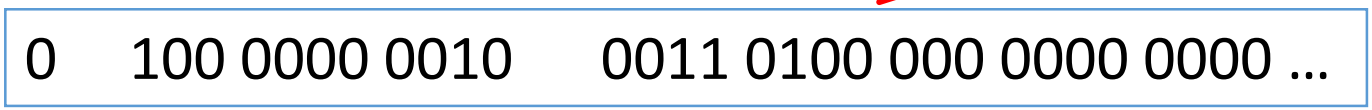
Formato IEEE754 64 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 64 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 1023_{10} = 1026_{10} \equiv 100\ 0000\ 0010_2$
- Mantisa: $1{,}001101_2$



Formato IEEE754 64 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 64 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 1023_{10} = 1026_{10} \equiv 100\ 0000\ 0010_2$
- Mantisa: $1{,}001101_2$



Formato IEEE754 64 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 64 bits:
  - $9{,}625_{10} \rightarrow 1{,}001101_2 * 2^3$
- Signo: 0
- Exponente: $3_{10} + 1023_{10} = 1026_{10} \equiv 100\ 0000\ 0010_2$
- Mantisa: $1{,}001101_2$

0100 0000 0010 0011 0100 000 0000 0000 …

4023400000000000

Formato IEEE754 64 bits

- Expresa el número $9{,}625_{10}$ en binario usando IEEE754 de 64 bits:

Solución: $9{,}625_{10} \equiv 4023400000000000_{16}$

De IEEE754 a decimal

IEEE754 a decimal

- Expresa el número $447A0800_{16}$ en decimal (el número está codificado en IEEE754):

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

1100 0100 0111 1010 0000 1000 0000 0000

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

1	100 0100 0	111 1010 0000 1000 0000 0000

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

1	100 0100 0	111 1010 0000 1000 0000 0000

Signo: negativo

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

1	100 0100 0	111 1010 0000 1000 0000 0000

Exponente: $100\ 0100\ 0_2 = 1000\ 1000_2 \equiv 136_{10}$

$$136_{10} - 127_{10} = 9_{10}$$

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

1	100 0100 0	111 1010 0000 1000 0000 0000

Mantisa: 1, 111 1010 0000 1000 0000 0000

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

$$C47A0800_{16} \rightarrow -1,111101000001_2 * 2^9$$

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

$$C47A0800_{16} \rightarrow -1,111101000001_2 * 2^9$$

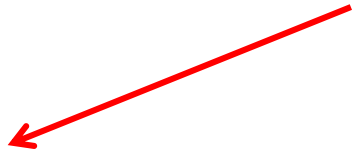
$$C47A0800_{16} \equiv -1111101000,001_2$$

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

$$C47A0800_{16} \rightarrow -1,111101000001_2 * 2^9$$

$$C47A0800_{16} \equiv -1111101000,001_2$$



$$1111101000_2 \equiv 1000_{10}$$

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

$$C47A0800_{16} \rightarrow -1,111101000001_2 * 2^9$$

$$C47A0800_{16} \equiv -1111101000,001_2$$

$$1111101000_2 \equiv 1000_{10} \qquad 0,001_2 \equiv 0,125_{10}$$

IEEE754 a decimal

- Expresa el número $C47A0800_{16}$ en decimal (el número está codificado en IEEE754):

$$C47A0800_{16} \equiv -1000{,}125_{10}$$

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

0100 0011 0000 0111 1111 0011 0011 011

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

0	100 0011 0	000 0111 1111 0011 0011 011

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

0	100 0011 0	000 0111 1111 0011 0011 011

Signo: positivo

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

0	100 0011 0	000 0111 1111 0011 0011 011

Exponente: $100\ 0011\ 0_2 = 1000\ 0110_2 \equiv 134_{10}$

$$134_{10} - 127_{10} = 7_{10}$$

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

0	100 0011 0	000 0111 1111 0011 0011 011

Mantisa: 1, 000 0111 1111 0011 0011 011

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

$4307F333_{16} \rightarrow 1{,}00001111111001100111011_2 * 2^7$

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

$$4307F333_{16} \rightarrow 1{,}00001111111001100111011_2 * 2^7$$

$$4307F333_{16} \equiv 10000111{,}11110011001110 11_2$$

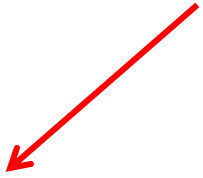
$$10000111_2 \equiv 135_{10}$$

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

$$4307F333_{16} \rightarrow 1{,}00001111111001100111011_2 * 2^7$$

$$4307F333_{16} \equiv 10000111{,}11110011001110 11_2$$



$$10000111_2 \equiv 135_{10} \qquad 0{,}11110011001110 11_2 \equiv 0{,}95_{10}$$

IEEE754 a decimal

- Expresa el número $4307F333_{16}$ en decimal (el número está codificado en IEEE754):

$$4307F333_{16} \equiv 135{,}95_{10}$$

Codificación de caracteres alfanuméricos

¿Qué es un carácter?

- “Buenos días”:
  - ‘B’
  - ‘u’
  - ‘e’
  - ‘n’
  - ‘o’
  - ‘s’
  - ‘ ‘
  - ‘d’
  - ‘í’
  - ‘a’
  - ‘s’

¿Qué es un carácter?

- “Buenos días”:
  - ‘B’
  - ‘u’
  - ‘e’
  - ‘n’
  - ‘o’
  - ‘s’
  - ‘ ‘
  - ‘d’
  - ‘í’
  - ‘a’
  - ‘s’

Es una cadena de caracteres
de 11 caracteres

ASCII

ASCII

- Primer intento
- Codifica el alfabeto latino para lengua **inglesa**

ASCII

- Cada carácter 1 byte

ASCII

- Cada carácter 1 byte
  - En realidad **7 bits**. El primer bit siempre a cero
  - 2^7= 128 caracteres

ASCII

- Cada carácter 1 byte
  - En realidad **7 bits**. El primer bit siempre a cero
  - 2^7= 128 caracteres
  - 33 caracteres de control
  - 95 caracteres imprimibles

ASCII

- Mayúsculas y minúsculas
  - Se diferencias en el tercer bit (32_{10})

  - ‘R’: 01**0**1 0010 (82_{10})
  - ‘r’: 01**1**1 0010 (114_{10})

ASCII

- Números
  - Los 4 últimos bits en binario (N=4)

  - ‘2’: 0011 0010
  - ‘8’: 0011 1000

ASCII

- http://es.wikipedia.org/wiki/ASCII

ISO latin 1

ISO latin 1

- Segundo intento
- Codifica la mayoría de lenguas de Europa occidental:
  - Castellano, Catalán, Alemán, Francés, etc.
- Incluye: Ñ, ¿, ç, etc.

ISO latin 1

- Cada carácter 1 byte
  - Usa los **8 bits**
  - 2^8 = 256 caracteres

ISO latin 1

- Los caracteres ASCII se codifican igual en ISO latin 1

ISO latin 1

- Hay muchos idiomas en la Tierra.
- Algunos (Chino) tienen muchos caracteres.
- 256 no es suficiente

ISO latin 1

- http://es.wikipedia.org/wiki/ISO_8859-1

UNICODE

UNICODE

- Tercer intento y definitivo
- Un código **único** para cada carácter posible.
- Incluye:
  - **todos** los idiomas, actuales y antiguos.
  - Símbolos: musicales, matemáticos, etc.

UNICODE

- http://www.unicode.org/charts/
- http://unicode-table.com/es/

UNICODE

- ATENCIÓN:

NO ES UNA CODIFICACIÓN

UTF-8

UTF-8

- Codificación de caracteres UNICODE.
- Sistema de longitud variable.

UTF-8

- UNICODE y UTF-8 no son lo mismo