Grupo ARCOS

uc3m | Universidad Carlos III de Madrid

Tema 2: Representación de la información

Estructura de Computadores

Grado en Ingeniería Informática
Grado en Matemática aplicada y Computación
Doble Grado en Ingeniería Informática y Administración de Empresas

Contenidos

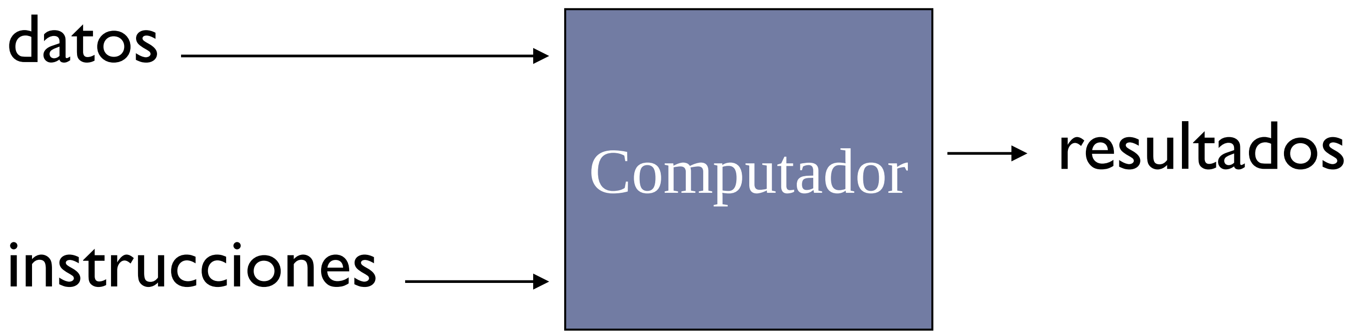
Contenidos

1. **Introducción**
   1. **Motivación y objetivos**
   2. **Sistemas posicionales**

2. Representaciones
   1. Alfanuméricas
      1. Caracteres
      2. Cadenas de caracteres
   2. Numéricas
      1. Naturales y enteras
      2. Coma fija
      3. Coma flotante (estándar IEEE 754)

Introducción: computador

- Un computador es una máquina destinada a procesar datos.



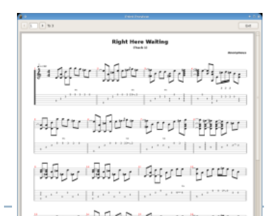
- Se aplican unas instrucciones y se obtiene unos resultados

Introducción:
computador

- Un computador es una máquina destinada a procesar datos.

datos → Computador → resultados
instrucciones → Computador

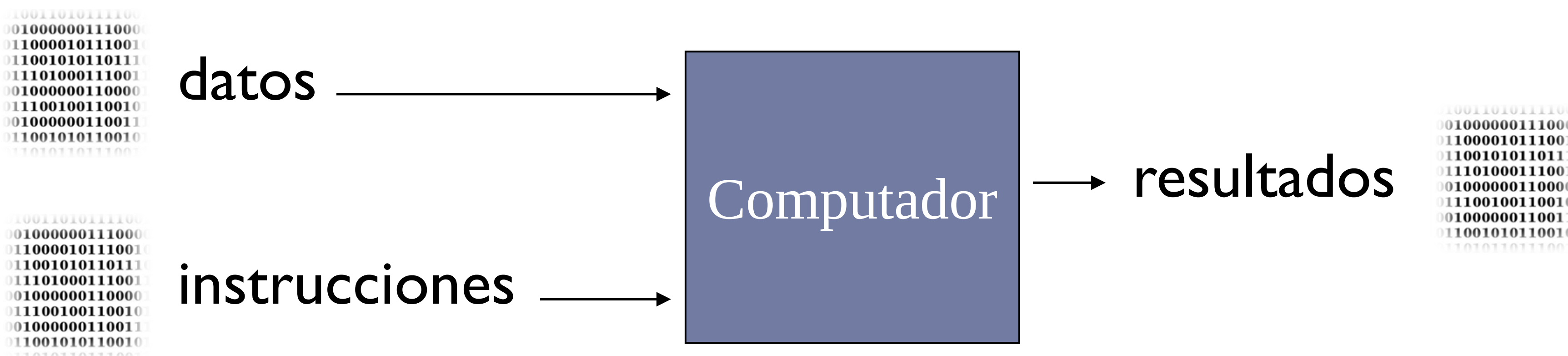
- Se aplican unas instrucciones y se obtiene unos resultados
- Los datos/información pueden ser de distintos tipos.

https://www.pixar.com/feature-films/up

Introducción:
computador

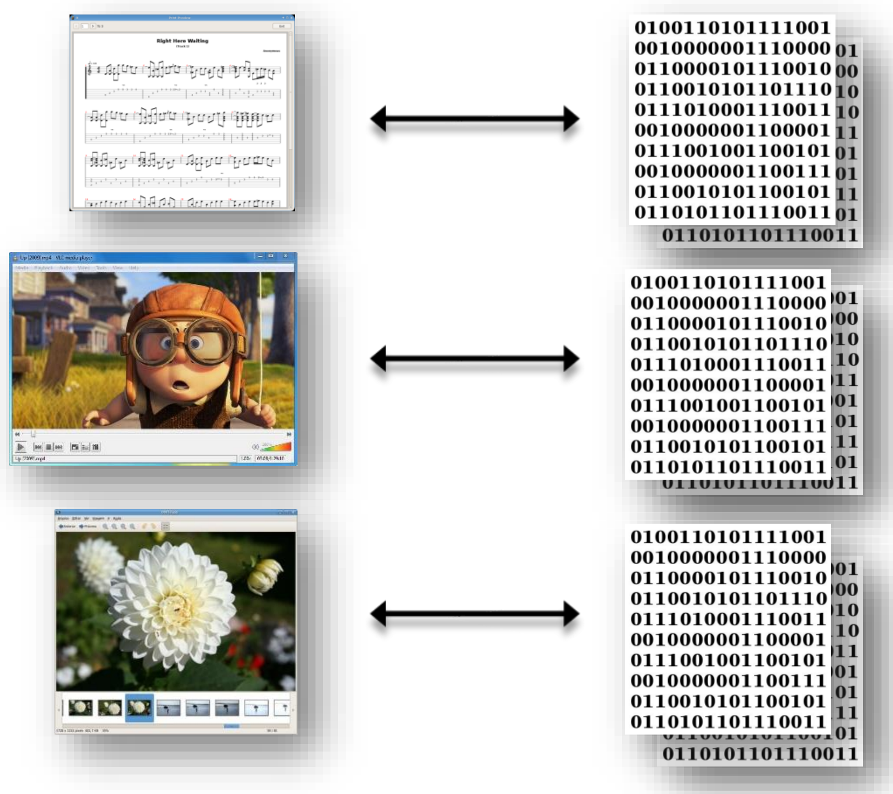
- Un computador es una máquina destinada a procesar datos.



  - Se aplican unas instrucciones y se obtiene unos resultados
  - Los datos/información pueden ser de distintos tipos.
  - Un computador solo usa una representación: binario.

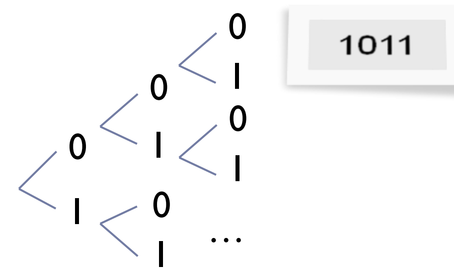
Introducción:
representación de la información

- El uso de una representación permite transformar los distintos tipos de información en binario (y viceversa)

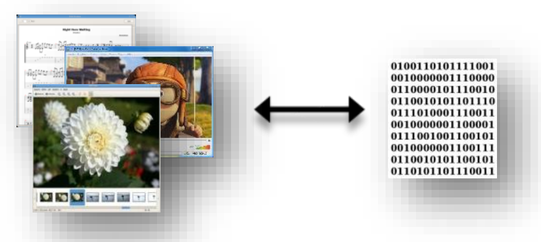


Introducción: características de la información

- Un computador maneja un conjunto finito de valores
  - Tipo binario (dos estados)
  - Finito (representación acotada)
    - Nº de bits de palabra del computador
      o bit (1), nibble (4), byte (8), half w., double w., …
    - Con **n** bits se pueden codificar **2^n** valores distintos



- Hay algunos tipos de información que son infinitos
  - Imposible representar todos los valores de los números naturales, reales, etc.

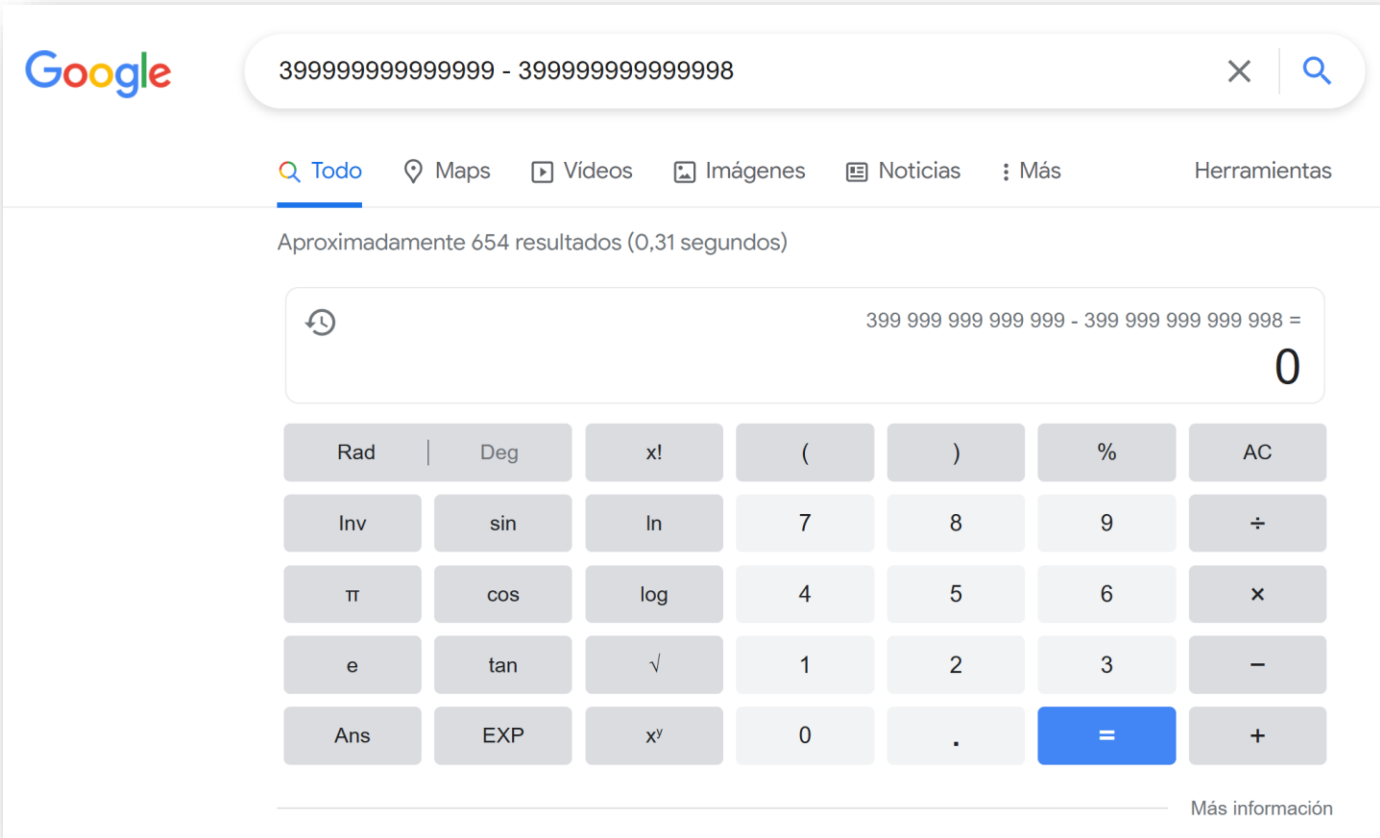


- La representación elegida tiene limitaciones

https://www.pixar.com/feature-films/up

Félix García-Carballeira, Alejandro Calderón Mateos

Ejemplo 1: la calculadora de Google con 15 dígitos…

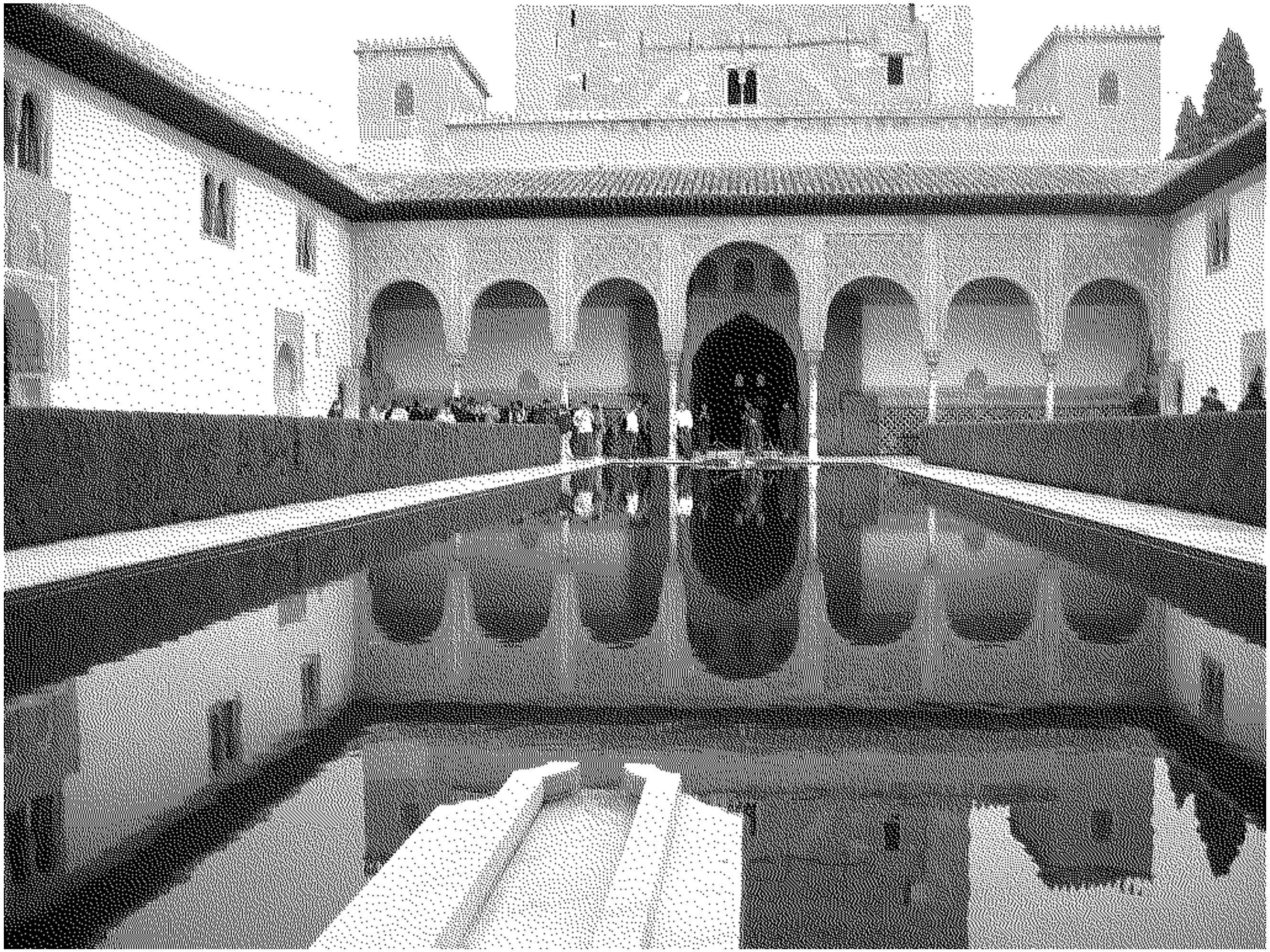


http://www.20minutos.es/noticia/415383/0/google/restar/error/

Félix García-Carballeira, Alejandro Calderón Mateos

Ejemplo 2:
la profundidad de color…

1 bit	2 colores
4 bits	16 colores
8 bits	256 colores



http://platea.pntic.mec.es/~lgonzale/tic/imagen/conceptos.html

Ejemplo 2:
la profundidad de color…

1 bit	2 colores
4 bits	16 colores
8 bits	256 colores

http://platea.pntic.mec.es/~lgonzale/tic/imagen/conceptos.html

Félix García-Carballeira, Alejandro Calderón Mateos

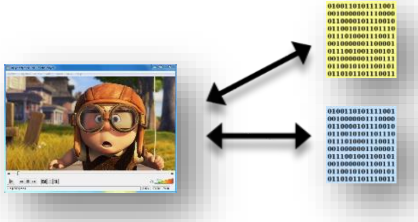
Ejemplo 2:
la profundidad de color…

1 bit	2 colores
4 bits	16 colores
8 bits	256 colores

http://platea.pntic.mec.es/~lgonzale/tic/imagen/conceptos.html

Necesitaremos…

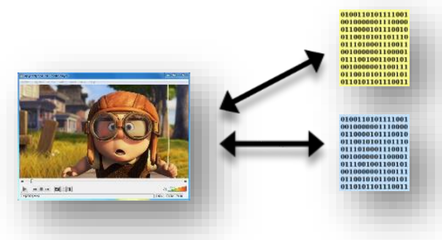
- Conocer posibles representaciones:



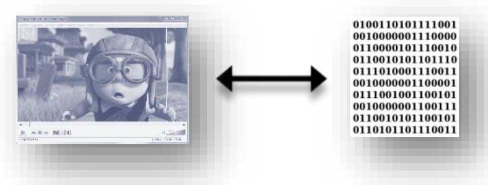
https://www.pixar.com/feature-films/up

Necesitaremos…

- Conocer posibles representaciones:



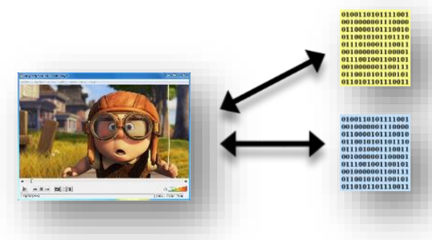
- Conocer las características de las mismas:
  - Limitaciones



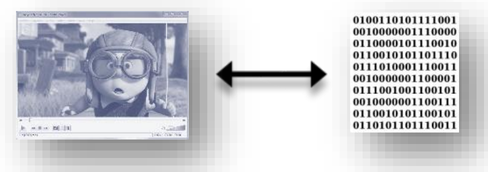
https://www.pixar.com/feature-films/up

Necesitaremos…

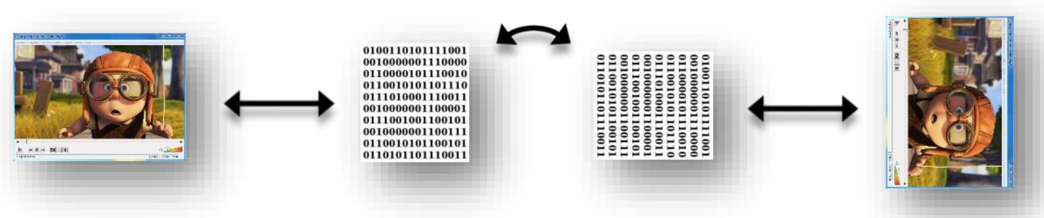
- Conocer posibles representaciones:



- Conocer las características de las mismas:
  - Limitaciones



- Conocer cómo operar con la representación:



https://www.pixar.com/feature-films/up

Contenidos

1. **Introducción**
   1. Motivación y objetivos
   2. **Sistemas posicionales**

2. Representaciones
   1. Alfanuméricas
      1. Caracteres
      2. Cadenas de caracteres
   2. Numéricas
      1. Naturales y enteras
      2. Coma fija
      3. Coma flotante (estándar IEEE 754)

Sistemas de representación posicionales

- Un número se define por una cadena de dígitos, estando afectado cada uno de ellos por un factor de escala que depende de la posición que ocupa en la cadena.

- Dada una base de numeración b, un número X se define como la cadena de dígitos:
  $X = (\ldots\ x_2\ x_1\ x_0\ ,\ x_{-1}\ x_{-2}\ \ldots)_b$ Con $0 \le x_i < b$
  con una lista de pesos asociados:
  $P = (\ldots\ b^2\ b^1\ b^0\ \ b^{-1}\ b^{-2}\ \ldots)_b$

- Su valor es:

$$V(X) = \sum_{i=-\infty}^{+\infty} b^i \cdot x_i = \cdots b^2 \cdot x_2 + b^1 \cdot x_1 + b^0 \cdot x_0 + b^{-1} \cdot x_{-1} + b^{-2} \cdot x_{-2} \cdots$$

Sistemas de representación posicionales

- Decimal

  X = 9 7 3 1

  … 10^3 10^2 10^1 10^0

- Binario

  X = 0 1 0 1

  … 2^3 2^2 2^1 2^0

- Hexadecimal

  X = B A 5 E

  … 16^3 16^2 16^1 16^0

Félix García-Carballeira, Alejandro Calderón Mateos

Sistemas de representación posicionales

- Decimal

  X = 9 7 3 1

  … 10^3 10^2 10^1 10^0

- Binario

  X = 0 1 0 1

  … 2^3 2^2 2^1 2^0

- Hexadecimal

  X = B A 5 E

  … 16^3 16^2 16^1 16^0

Paso de binario a hexadecimal:

- Agrupar de 4 en 4 bits, de derecha a izquierda
- Cada 4 bits es el valor del dígito hexadecimal
- Ej.: $\underbrace{1\ 0\ 1\ 0}_{A}\ \underbrace{0\ 1\ 0\ 1}_{5}$

  0x A 5

Félix García-Carballeira, Alejandro Calderón Mateos

Sistemas de representación posicionales

- Decimal

  X = 9 7 3 1

  ... 10^3 10^2 10^1 10^0

¿?

- Binario

  X = 0 1 0 1

  ... 2^3 2^2 2^1 2^0

- Hexadecimal

  X = B A 5 E

  ... 16^3 16^2 16^1 16^0

Ejercicio

- Representar 342 en binario:

256	128	64	32	16	8	4	2	1
?	?	?	?	?	?	?	?	?

Ejercicio (solución)

- Representar 342 en binario:

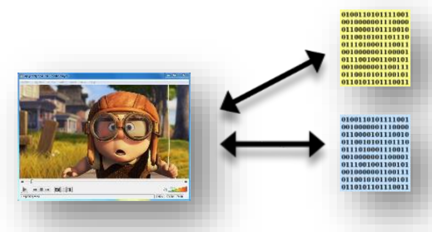
256	128	64	32	16	8	4	2	1
1	0	1	0	1	0	1	1	0

342-256=86 86-64=22 22-16=6 6-4=2 2-2=0

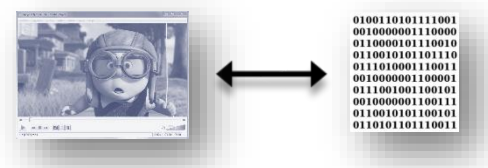
Necesitaremos…

Recordatorio

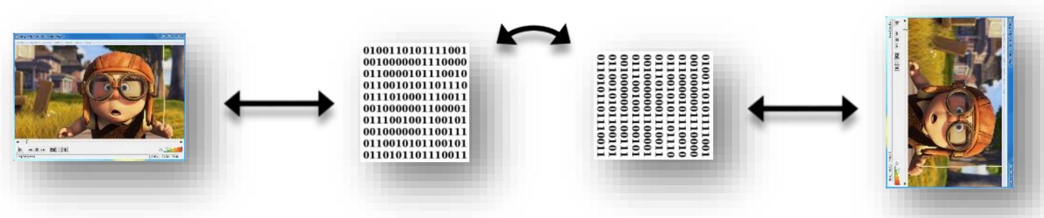
- Conocer posibles representaciones:



- Conocer las características de las mismas:
  - Limitaciones



- Conocer cómo operar con la representación:

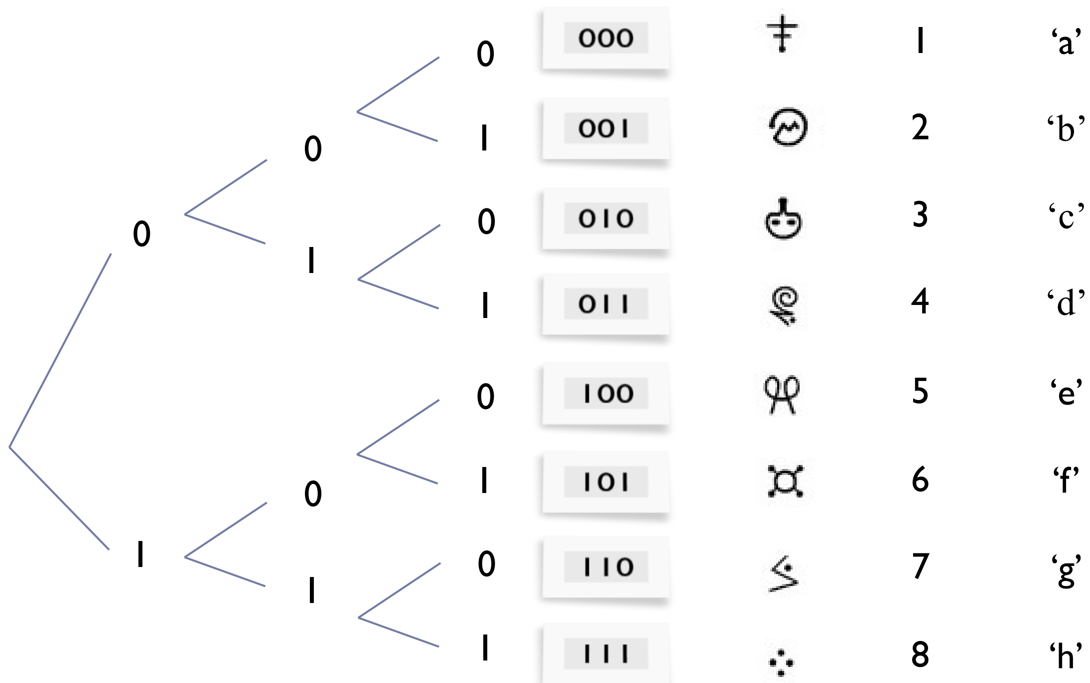


https://www.pixar.com/feature-films/up

Félix García-Carballeira, Alejandro Calderón Mateos

Sistemas de representación posicionales

- Con 3 dígitos binarios, se pueden representar 8 símbolos:



Sistemas de representación posicionales

- ¿Cuántos valores se pueden representar con n bits?

- ¿Cuántos bits se necesitan para representar m 'valores'?

- Con n bits,
  si el valor mínimo representable corresponde al número 0,
  ¿Cuál es el máximo valor numérico representable?

Sistemas de representación posicionales

- ¿Cuántos valores se pueden representar con n bits?
  - **2^n**

  - Ej.: con 4 bits se pueden representar 16 valores

- ¿Cuántos bits se necesitan para representar m 'valores'?
  - $\lceil$**Log2(n)**$\rceil$ ($Log_2(n)$ por exceso)
  - Ej.: para representar 35 valores se necesitan 6 bits

- Con n bits,
  si el valor mínimo representable corresponde al número 0,
  ¿Cuál es el máximo valor numérico representable?
  - **2^n-1**

Ejercicio

- Calcular el valor de (23 unos):

$$11111111111111111111111_2$$

Félix García-Carballeira, Alejandro Calderón Mateos

Ejercicio (solución)

- Calcular el valor de (23 unos):

$$11111111111111111111111_2$$

$$\mathbf{X = 2^{23} - 1}$$

Truco:

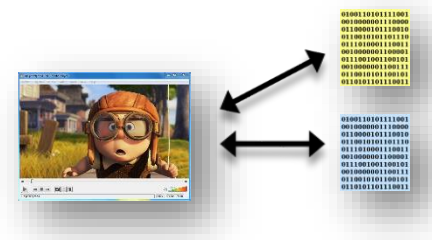
$$
\begin{array}{rl}
11111111111111111111_2 & = X \\
+ \;\; 00000000000000000001_2 & = 1 \\
\hline
100000000000000000000_2 & = 2^{23}
\end{array}
$$

$$X = 2^{23} - 1$$

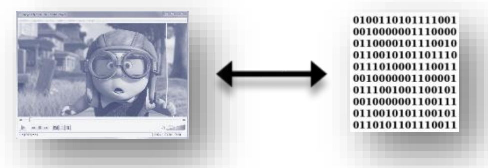
Necesitaremos…

Recordatorio

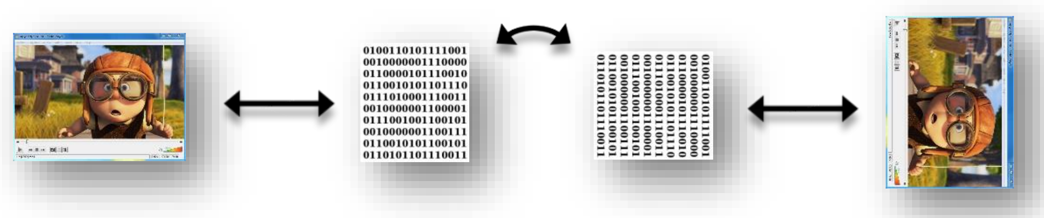
- Conocer posibles representaciones:



- Conocer las características de las mismas:
  - Limitaciones



- Conocer cómo operar con la representación:



https://www.pixar.com/feature-films/up

Operaciones con representación binaria

- Sumar en binario:

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

Félix García-Carballeira, Alejandro Calderón Mateos

Operaciones con representación binaria

- **Sumar** en binario:

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

- **Restar** en binario:

$$
\begin{array}{cccccc}
 & & & 1 & 1 & \\
 & 0 & 1 & 1 & 0 & 0 \\
- & 0 & 1 & 0 & 1 & 1 \\
\hline
 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

Félix García-Carballeira, Alejandro Calderón Mateos

Ejercicio

2 minutos máx.

Tienes una botella de 5 litros y otra de 3 litros. ¿Cómo conseguir justo 4 litros de agua?

http://www.educaplanet.com/educaplanet/2014/11/acertijo-logica-las-jarras-de-agua/

Ejercicio (solución)

2 minutos máx.

Tienes una botella de 5 litros y otra de 3 litros. ¿Cómo conseguir justo 4 litros de agua?

- Llene la jarra de 5 litros
- Vacíalo en la jarra de 3 litros
  - Quedan 2 en la jarra de 5 litros (-3 a 5).
- Tira lo que hay en la jarra de 3 litros
- Transfiere los 2 de la jarra de 5 litros a la de 3 litros
  - Queda 1 en la jarra de 3 litros (-1 a 3).
- Rellena la jarra de 5 litros
- Llena la jarra de 3 litros hasta arriba, lo que queda en la jarra de 5 litros son 4 litros

http://www.educaplanet.com/educaplanet/2014/11/acertijo-logica-las-jarras-de-agua/

Ejercicio

2 minutos máx.

- Utilizando los números 112 y -71 en base decimal, realiza la suma en complemento a 10.

Ejercicio (solución)

2 minutos máx.

- Complemento a 10 de 112 es: 112
- Complemento a 10 de -71 es:

```
 1000
-0071
--------
  929
```

- Sumando ambos:

```
  112
  929
--------
~~1~~041
```

```
  112
 -071
--------
  041
```


Félix García-Carballeira, Alejandro Calderón Mateos

Ejercicio

2 minutos máx.

- Utilizando los números 110 y -011 en binario, realiza la suma en complemento a 2.

Ejercicio (solución)

2 minutos máx.

- Complemento a 2 de 110 es: 110
- Complemento a 2 de -011 es:

  1000
  -011

  100+1

- Sumando ambos:

  110
  101

  ~~1~~011

  110
  -011

  011

Félix García-Carballeira, Alejandro Calderón Mateos

Contenidos

1. Introducción
   1. Motivación y objetivos
   2. Sistemas posicionales
2. **Representaciones**
   1. **Alfanuméricas**
      1. **Caracteres**
      2. **Cadenas de caracteres**
   2. Numéricas
      1. Naturales y enteras
      2. Coma fija
      3. Coma flotante (estándar IEEE 754)

Representación alfanumérica

- Cada carácter se codifica con un byte.
- Para n bits $\Rightarrow 2^n$ caracteres representables:

# bits	# caracteres	Incluye…	Ejemplo
6	64	• 26 letras: a...z • 10 números: 0...9 • Puntuación: . , ; : ... • Especiales: + - [...	***BCDIC***
7	128	• añade mayúsculas y caracteres de control	***ASCII***
8	256	• añade letras acentuadas, ñ, caracteres semigráficos	***EBCDIC*** ***ASCII extendido***
16	34.168	• Añade distintos idiomas (chino, árabe,...)	***UNICODE***

Ejemplo: tabla ASCII (7 bits)

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	¦
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

Ejemplo: tabla ASCII (7 bits)

caracteres de control

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	‼	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	¦
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

< 32

Ejemplo: tabla ASCII (7 bits)
distancia mayúsculas-minúsculas

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	`
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	¦
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

97-65=32

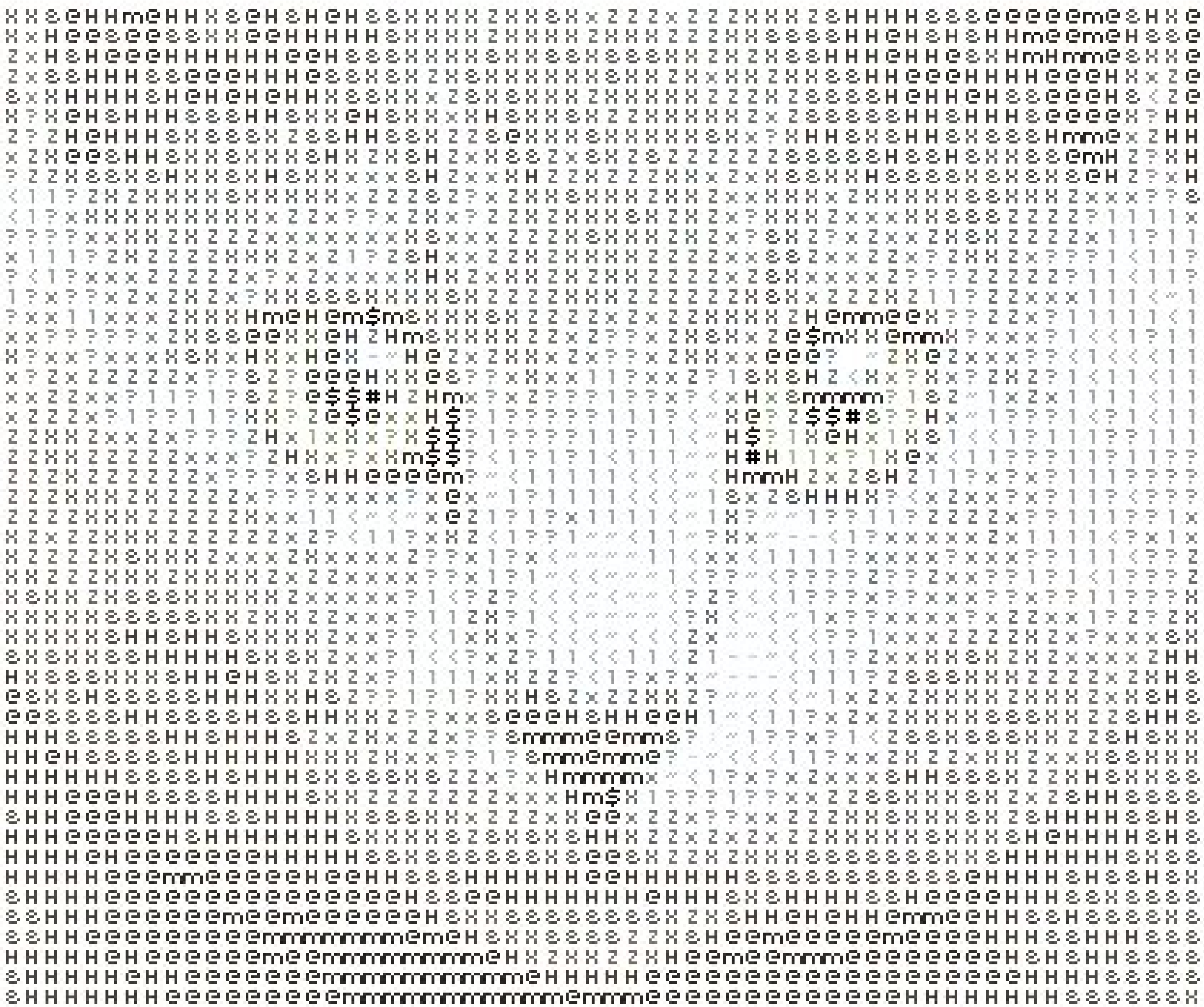
Ejemplo: tabla ASCII (7 bits)

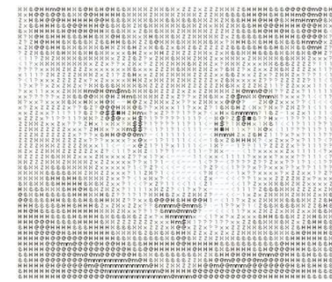
conversión de un número a carácter

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	¦
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	∧	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

6+48=54

Curiosidad:
Visualización ‘gráfica’ con caracteres





http://www.typorganism.com/asciiomatic/

Cadenas de caracteres

1000	00110011
1001	01101100
	…
1008	10100011

1. Cadenas de longitud fija:

h	o	l	a				
1	2	3	4	5	6	7	8

2. Cadenas de longitud variable con separador:

h	o	l	a	\0			
1	2	3	4	5	6	7	8

3. Cadenas de longitud variable con longitud en cabecera:

4	h	o	l	a			
1	2	3	4	5	6	7	8

Contenidos

1. Introducción
   1. Motivación y objetivos
   2. Sistemas posicionales

2. Representaciones
   1. Alfanuméricas
      1. Caracteres
      2. Cadenas de caracteres
   2. **Numéricas**
      1. **Naturales y enteras**
      2. Coma fija
      3. Coma flotante (estándar IEEE 754)

Félix García-Carballeira, Alejandro Calderón Mateos

Representación numérica

- Clasificación de números reales:
  - Naturales: 0, 1, 2, 3, ...
  - Enteros: ... -3, -2, -1, 0, 1, 2, 3, ...
  - Racionales: fracciones (5/2 = 2,5)
  - Irracionales: $2^{1/2}$, π, e, ...

- Conjuntos infinitos y espacio de representación finito:
  - Imposible representar todos

- Características de la representación usada:
  - Elemento representado:
    Natural, entero, …
  - Rango de representación:
    Intervalo entre el menor y mayor nº representable
  - Resolución de representación:
    Diferencia entre un nº representable y el siguiente.
    Representa el máximo error cometido. Puede ser cte. o variable.

Sistemas de representación binarios más usados

A. Coma fija sin signo o binario puro — naturales

B. Signo magnitud

C. Complemento a uno (Ca 1)

D. Complemento a dos (Ca 2) — enteros

E. Exceso $2^{n-1}-1$

F. Coma flotante: Estándar IEEE 754 — reales

Félix García-Carballeira, Alejandro Calderón Mateos

n-1 0

n bits

- Rango de representación: $[0, 2^n - 1]$
- Resolución: 1 unidad

Decimal	Binario Puro	Signo magnitud	Complemento a uno	Complemento a dos	Exceso 3
+7	111	N.D.	N.D.	N.D.	N.D.
+6	110	N.D.	N.D.	N.D.	N.D.
+5	101	N.D.	N.D.	N.D.	N.D.
+4	100	N.D.	N.D.	N.D.	111
+3	011	011	011	011	110
+2	010	010	010	010	101
+1	001	001	001	001	100
+0	000	000	000	000	011
-0	N.D.	100	111	N.D.	N.D.
-1	N.D.	101	110	111	010
-2	N.D.	110	101	110	001
-3	N.D.	111	100	101	000
-4	N.D.	N.D.	N.D.	100	N.D.
-5	N.D.	N.D.	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.	N.D.	N.D.

Félix García-Carballeira, Alejandro Calderón Mateos

n-1, n-2 0

s	n bits

$0 \Rightarrow +$; $1 \Rightarrow -$

- Rango de representación: $[-2^{n-1}+1, \ 2^{n-1}-1]$
- Resolución: 1 unidad
- Ambigüedad del 0 (+0 y -0)

Decimal	Binario Puro	Signo magnitud	Complemento a uno	Complemento a dos	Exceso 3
+7	111	N.D.	N.D.	N.D.	N.D.
+6	110	N.D.	N.D.	N.D.	N.D.
+5	101	N.D.	N.D.	N.D.	N.D.
+4	100	N.D.	N.D.	N.D.	111
+3	011	011	011	011	110
+2	010	010	010	010	101
+1	001	001	001	001	100
+0	000	000	000	000	011
-0	N.D.	100	111	N.D.	N.D.
-1	N.D.	101	110	111	010
-2	N.D.	110	101	110	001
-3	N.D.	111	100	101	000
-4	N.D.	N.D.	N.D.	100	N.D.
-5	N.D.	N.D.	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.	N.D.	N.D.

Ejemplo

- ¿Se puede representar 745_{10} en signo magnitud con 10 bits?

Ejemplo (solución)

- ¿Se puede representar 745_{10} en signo magnitud con 10 bits?

- Con 10 bits el rango en signo magnitud es: $[-2^9+1,\ldots,-0,+0,\ldots 2^9-1] \Rightarrow [-511, 511]$ y por tanto, **no podemos representar** 745

n-1, n-2 0

s	n bits

0 ⇒ + ; 1 ⇒ - y parte de magnitud

- Rango de representación: $[-2^{n-1}+1, \ 2^{n-1}-1]$
- Resolución: 1 unidad
- Ambigüedad del 0 / rango simétrico
- -: 2^n-X-1 o cambiar 1 por 0 y 0 por 1

Decimal	Binario Puro	Signo magnitud	Complemento a uno	Complemento a dos	Exceso 3
+7	111	N.D.	N.D.	N.D.	N.D.
+6	110	N.D.	N.D.	N.D.	N.D.
+5	101	N.D.	N.D.	N.D.	N.D.
+4	100	N.D.	N.D.	N.D.	111
+3	011	011	011	011	110
+2	010	010	010	010	101
+1	001	001	001	001	100
+0	000	000	000	000	011
-0	N.D.	100	111	N.D.	N.D.
-1	N.D.	101	110	111	010
-2	N.D.	110	101	110	001
-3	N.D.	111	100	101	000
-4	N.D.	N.D.	N.D.	100	N.D.
-5	N.D.	N.D.	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.	N.D.	N.D.

Ejemplo

Para n = 5 bits y usando complemento a uno:

- ¿Cómo se representa X = 5?
- ¿Cómo se representa X = -5?
- ¿Cuál es el valor de 00111 en complemento a 1?
- ¿Cuál es el valor de 11000 en complemento a 1?

Félix García-Carballeira, Alejandro Calderón Mateos

Ejemplo (solución)

Para n = 5 bits y usando complemento a uno:

- ¿Cómo se representa X = 5?
  - Como es positivo, en binario puro
    - 00101
- ¿Cómo se representa X = -5?
  - Como es negativo, se complementa el valor 5 (00101)
    - 11010
- ¿Cuál es el valor de 00111 en complemento a 1?
  - Como es positivo, su valor es directamente 7
- ¿Cuál es el valor de 11000 en complemento a 1?
  - Como es negativo, se complementa y se obtiene 00111 (7)
    - El valor es -7

n-1, n-2 0

s	n bits

0 ⇒ + ; 1 ⇒ - y parte de magnitud

- Rango de representación: $[-2^{n-1},\ 2^{n-1}-1]$
- Resolución: 1 unidad
- No ∃ -0 / rango **asimétrico**
- -: 2^n-X o complemento a uno más uno

Decimal	Binario Puro	Signo magnitud	Complemento a uno	Complemento a dos	Exceso 3
+7	111	N.D.	N.D.	N.D.	N.D.
+6	110	N.D.	N.D.	N.D.	N.D.
+5	101	N.D.	N.D.	N.D.	N.D.
+4	100	N.D.	N.D.	N.D.	111
+3	011	011	011	011	110
+2	010	010	010	010	101
+1	001	001	001	001	100
+0	000	000	000	000	011
-0	N.D.	100	111	N.D.	N.D.
-1	N.D.	101	110	111	010
-2	N.D.	110	101	110	001
-3	N.D.	111	100	101	000
-4	N.D.	N.D.	N.D.	100	N.D.
-5	N.D.	N.D.	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.	N.D.	N.D.

Complemento a dos para 32 bits

$0000 \ldots 0000\ 0000\ 0000\ 0000_{dos} = 0_{(10}$
$0000 \ldots 0000\ 0000\ 0000\ 0001_{dos} = 1_{(10}$
$0000 \ldots 0000\ 0000\ 0000\ 0010_{dos} = 2_{(10}$
. . .
$0111 \ldots 1111\ 1111\ 1111\ 1101_{dos} = 2{,}147{,}483{,}645_{(10}$
$0111 \ldots 1111\ 1111\ 1111\ 1110_{dos} = 2{,}147{,}483{,}646_{(10}$
$0111 \ldots 1111\ 1111\ 1111\ 1111_{dos} = 2{,}147{,}483{,}647_{(10}$

$1000 \ldots 0000\ 0000\ 0000\ 0000_{dos} = -2{,}147{,}483{,}648_{(10}$
$1000 \ldots 0000\ 0000\ 0000\ 0001_{dos} = -2{,}147{,}483{,}647_{(10}$
$1000 \ldots 0000\ 0000\ 0000\ 0010_{dos} = -2{,}147{,}483{,}646_{(10}$
. . .
$1111 \ldots 1111\ 1111\ 1111\ 1101_{dos} = -3_{(10}$
$1111 \ldots 1111\ 1111\ 1111\ 1110_{dos} = -2_{(10}$
$1111 \ldots 1111\ 1111\ 1111\ 1111_{dos} = -1_{(10}$

n-1, n-2 0

s	n bits

0 ⇒ + ; 1 ⇒ - y parte de magnitud

- Rango de representación: $[-2^{n-1}+1,\ 2^{n-1}]$
- Resolución: 1 unidad
- No ∃ -0 / rango asimétrico
- +/-: X + 2^{n-1} (sesgo = 2^{n-1} - 1)

Decimal	Binario Puro	Signo magnitud	Complemento a uno	Complemento a dos	Exceso 3
+7	111	N.D.	N.D.	N.D.	N.D.
+6	110	N.D.	N.D.	N.D.	N.D.
+5	101	N.D.	N.D.	N.D.	N.D.
+4	100	N.D.	N.D.	N.D.	111
+3	011	011	011	011	110
+2	010	010	010	010	101
+1	001	001	001	001	100
+0	000	000	000	000	011
-0	N.D.	100	111	N.D.	N.D.
-1	N.D.	101	110	111	010
-2	N.D.	110	101	110	001
-3	N.D.	111	100	101	000
-4	N.D.	N.D.	N.D.	100	N.D.
-5	N.D.	N.D.	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.	N.D.	N.D.

Félix García-Carballeira, Alejandro Calderón Mateos

Representaciones

resumen

Nombre	**Binario puro**	**Signo-magnitud**	**Ca1**	**Ca2**	**Exceso 2^{n-1}-1**
Representa	Natural	Entero	Entero	Entero	Entero
Signo	Todos los bits son magnitud, no hay signo	MSB es el signo (0 ⇒ + y 1 ⇒ **-**)	MSB es signo **y magnitud** (0 ⇒ + y 1 ⇒ **-**)	MSB es signo **y magnitud** (0 ⇒ + y 1 ⇒ **-**)	MSB es signo **y magnitud** (0 ⇒ **- o 0** y 1 ⇒ **+ no 0**)
Rango	$[0, 2^n-1]$	$[-2^{n-1}+1,\ 2^{n-1}-1]$	$[-2^{n-1}+1, 2^{n-1}-1]$	$[-2^{n-1}, 2^{n-1}-1]$	$[-(2^{n-1}-1), 2^{n-1}]$
Resolución	1 unidad	1 unidad	1 unidad	1 unidad	1 unidad
Inconveniente	No negativos	+0 y -0	+0 y -0	Rango **a**simétrico	Rango **a**simétrico
Ventaja		Rango simétrico	Rango simétrico	(No ∃ **-**0)	(No ∃ **-**0)
Truco		Quitar primer bit y con el resto es igual que binario	+: = binario -: cambiar 1 por 0 y 0 por 1	+: = binario -: Ca1 + 1	Restar siempre el sesgo (2^{n-1}-1)
Valor		$V(X) = (1 - 2 \cdot x_{n-1}) \cdot \sum_{i=0}^{n-2} 2^i \cdot x_i$	+: $V(X) = \sum_{i=0}^{n-2} 2^i \cdot x_i$ -: $V(X) = -2^n + \sum_{i=0}^{n-1} 2^i \cdot X_i + 1$	+: $V(X) = \sum_{i=0}^{n-2} 2^i \cdot x_i$ -: $V(X) = -2^n + \sum_{i=0}^{n-1} 2^i \cdot X_i$	$V(X) = \sum_{i=0}^{n-1} 2^i \cdot x_i - (2^{n-1} - 1)$

Félix García-Carballeira, Alejandro Calderón Mateos

Ejemplo comparativo (3 bits)

resumen

Decimal	Binario Puro	Signo magnitud	Complemento a uno	Complemento a dos	Exceso 3
+7	111	N.D.	N.D.	N.D.	N.D.
+6	110	N.D.	N.D.	N.D.	N.D.
+5	101	N.D.	N.D.	N.D.	N.D.
+4	100	N.D.	N.D.	N.D.	111
+3	011	011	011	011	110
+2	010	010	010	010	101
+1	001	001	001	001	100
+0	000	000	000	000	011
-0	N.D.	100	111	N.D.	N.D.
-1	N.D.	101	110	111	010
-2	N.D.	110	101	110	001
-3	N.D.	111	100	101	000
-4	N.D.	N.D.	N.D.	100	N.D.
-5	N.D.	N.D.	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.	N.D.	N.D.

Félix García-Carballeira, Alejandro Calderón Mateos

Ejercicio

Indique la representación de los siguientes números, razonando brevemente su respuesta:

1. **-32** en complemento a uno con **6 bits**
2. **-32** en complemento a dos con **6 bits**
3. **-10** en signo magnitud con **5 bits**
4. **+14** en complemento a dos con **5 bits**

Ejercicio (solución)

1. Con 6 bits **no es representable** en C1: $[-2^{6-1}+1,\ldots,-0,+0,\ldots 2^{6-1}-1]$

2. C1 + 1 -> **100000**

3. Signo=1, magnitud=1010 -> **11010**

4. Positivo -> C1=C2=SM -> **01110**

Contenidos

1. Introducción
    1. Motivación y objetivos
    2. Sistemas posicionales
2. Representaciones
    1. Alfanuméricas
        1. Caracteres
        2. Cadenas de caracteres
    2. **Numéricas**
        1. **Naturales y enteras**
            1. **Operaciones aritméticas**
        2. Coma fija
        3. Coma flotante (estándar IEEE 754)

Comparación de aritmética en BP, C1 y C2

	Binario puro	Complemento a 1	Complemento a 2
Suma	10110 01100 ---------- 100010	igual que B.P.	igual que B.P.
Resta	10110 01100 ---------- 01010	sumar y si hay Cn-1 entonces sumar Cn-1 al total	sumar y si hay Cn-1 entonces descartarlo

En hardware, es más fácil operar con complemento

Comparación de aritmética en BP, C1 y C2

por qué sumar el acarreo en Ca1

	Binari		a 2
Suma			
Resta	10110 01100 ---------- 01010	sumar y si hay Cn-1 entonces sumar Cn-1 al total	sumar y si hay Cn-1 entonces descartarlo

- -X se representa como $2^n - X \quad - 1$
- -Y se representa como $2^n - Y \quad - 1$
- -(X + Y) se representa como $2^n - (X+Y) - 1$

- -(X + Y) operando resulta $2^n + 2^n - (X + Y) - 2 \;\; \boxed{+1}$

Corrección de resultado sumando el acarreo…

Félix García-Carballeira, Alejandro Calderón Mateos

Comparación de aritmética en BP, C1 y C2

por qué descartar el acarreo en Ca2

	Binari		a 2
Suma			
Resta	10110 01100 ---------- 01010	sumar y si hay Cn-1 entonces sumar Cn-1 al total	sumar y si hay Cn-1 entonces descartarlo

- -X se representa como $2^n - X$
- -Y se representa como $2^n - Y$
- -(X + Y) se representa como $2^n - (X+Y)$

- -(X + Y) operando resulta $2^n + 2^n - (X + Y)$

Corrección de resultado descartando el acarreo…

Félix García-Carballeira, Alejandro Calderón Mateos

Comparación de aritmética en BP, C1 y C2

	Binario puro	Complemento a 1	Complemento a 2
Detectar desbordamiento	El resultado necesita 1 bit más Hay Cn	Suma de + + es –, Suma de – – es + Cn <> Cn-1	Suma de + + es –, Suma de – – es + Cn <> Cn-1
Extensión de signo	0…0 10110	1…1 10110 0…0 00110	1…1 10110 0…0 00110
…	…	…	…

Extensión de signo en complemento a dos

- ¿Cómo pasar de n bits a m bits, siendo n < m?
- Ejemplo:
  - n = 4, m = 8
  - Si X = 0110 con 4 bits $\Rightarrow$ X = 00000110 con 8 bits
  - Si X = 1011 con 4 bits $\Rightarrow$ X = 11111011 con 8 bits

Ejercicio

- Usando 5 bits para representarlo, haga las siguientes sumas en complemento a uno:
  a) 4 +12
  b) 4 -12
  c) -4 -12

Félix García-Carballeira, Alejandro Calderón Mateos

Ejercicio (Solución Ca1 con 5 bits)

a) 4 +12

```
00100
01100
--------
10000
```
10000 ⇒ se obtiene un negativo ⇒ -15 ⇒ overflow

b) 4 - 12

```
00100
10011
--------
10111
```
10111 ⇒ -8

c) -4 - 12

```
11011
10011
--------
101110
```
101110 ⇒ negativo con 6 bits ⇒ overflow

Félix García-Carballeira, Alejandro Calderón Mateos

Contenidos

Otras necesidades de representación

- ¿Cómo representar?
  - Números muy grandes: $30.556.926.000_{(10}$
  - Números muy pequeños: $0.0000000000529177_{(10}$
  - Números con decimales: 1,58567

Félix García-Carballeira, Alejandro Calderón Mateos

Ejemplo de fallo…

- Explosión del **Ariane 5** (primer viaje)
  - Enviado por ESA en junio de 1996
  - Coste del desarrollo:
    10 años y **7000 millones de dólares**
  - Explotó 40 segundos después de despegar, a 3700 metros de altura.
  - Fallo debido a la pérdida total de la información de altitud:
    - El software del sistema de referencia inercial realizó la conversión de un valor real en coma flotante de 64 bits a un valor entero de 16 bits. El número a almacenar era mayor de 32767 (el mayor entero con signo de 16 bits) y se produjo un fallo de conversión y una excepción.

http://www.rvs.uni-bielefeld.de/lectures/TechInf/TI2/download/ariane_pics.html

Coma fija [racionales]

- Se fija la posición de la coma binaria y se utilizan los pesos asociados a las posiciones decimales

- Ejemplo:

$$1001.1010 = 2^4 + 2^0 + 2^{-1} + 2^{-3} = 9{,}625$$

Representación de fracciones con representación binaria en coma fija

- Ejemplo de representación con 6 bits:

$$\mathtt{xx.yyyy}$$

$2^1 \quad 2^0 \quad 2^{-1} \quad 2^{-2} \quad 2^{-3} \quad 2^{-4}$

- Ejemplo de número:
  $10{,}1010_{(2} = 1\mathrm{x}2^1 + 1\mathrm{x}2^{-1} + 1\mathrm{x}2^{-3} = 2.625_{10}$
- Asumiendo esta coma fija, el rango sería:
  - [0 a 3.9375 (casi 4)]

Félix García-Carballeira, Alejandro Calderón Mateos

Potencias negativas

i	2^{-i}	
0	1.0	1
1	0.5	1/2
2	0.25	1/4
3	0.125	1/8
4	0.0625	1/16
5	0.03125	1/32
6	0.015625	
7	0.0078125	
8	0.00390625	
9	0.001953125	
10	0.0009765625	
11	…	

Contenidos

1. Introducción
   1. Motivación y objetivos
   2. Sistemas posicionales

2. Representaciones
   1. Alfanuméricas
      1. Caracteres
      2. Cadenas de caracteres
   2. **Numéricas**
      1. Naturales y enteras
      2. Coma fija
      3. **Coma flotante (estándar IEEE 754)**

Notación científica decimal

$$\underbrace{9.12}_{\text{mantisa}}{}_{10} \times \underbrace{10}_{\text{base}}{}^{\overset{\text{exponente}}{25}}$$

- Cada número lleva asociado una mantisa y un exponente
- Notación científica decimal usada: notación normalizada
  - Solo un dígito distinto de 0 a la izquierda del punto

- Se adapta el número al orden de magnitud del valor a representar, trasladando la *coma decimal* mediante el exponente

Notación científica en binario

mantisa

exponente

$$1.0_{(2}\ x\ 2^{-1}$$

coma binaria

base

- Forma normalizada: Un 1 (solo un dígito) a la izq. de la coma
  - Normalizada: 1.0001×2^{-9}
  - No normalizada: 0.0011×2^{-8}, 10.0×2^{-10}

Estándar IEEE 754-2008 [racionales]

1 0010…0101 00010…001

signo	**exponente**	**mantisa**
sign	exponent	significand

- Estándar para coma flotante usado en la mayoría de los ordenadores.
- **Características** (salvo casos especiales):
  - Exponente: en **exceso** con sesgo $2^{\text{num\_bits\_exponente} - 1} - 1$
  - Mantisa: **signo-magnitud**, **normalizada**, con **bit implícito**
- Diferentes **formatos**:
  - **Precisión simple**: 32 bits (signo: 1, exponente: 8 y mantisa: 23)
  - **Doble precisión**: 64 bits (signo: 1, exponente: 11 y mantisa: 52)
  - **Cuádruple precisión**: 128 bits (signo: 1, exponente: 15 y mantisa: 112)
  - **Octuple precisión**: 256 bits (signo: 1, exponente: 19 y mantisa: 237)

Normalización y bit implícito

- Normalización

  Para normalizar la mantisa se ajusta el exponente para que el bit más significativo de la mantisa sea 1

  - Ejemplo: 100100000000000000000000 x 2^3 (ya está normalizado)
  - Ejemplo: 000100000000010101 x 2^3 (no lo está)

    100000000010101000 x 2^0 (ahora sí)

- Bit implícito

  Una vez normalizado, dado que el bit más significativo es 1, **no** se almacena para dejar espacio para un bit más (aumenta la precisión)

  - Así se puede representar mantisas con un bit más

Félix García-Carballeira, Alejandro Calderón Mateos

Estándar IEEE 754 de precisión simple [racionales]

bits	0	1 … 8	9 … 31
	S	E	M

S es el signo (1 bit)
E es el exponente (8 bits)
M es la mantisa (23 bits)

- El valor se calcula con la siguiente expresión (salvo casos especiales):

$$\mathbf{N = (-1)^{S} \times 2^{E-127} \times 1.M}$$

donde:

S = 0 indica número positivo, S =1 indica número negativo
0 < E < 255 (E=0 y E=255 indican casos especiales)
$00000000000000000000000 \leq M \leq 11111111111111111111111$

Félix García-Carballeira, Alejandro Calderón Mateos

Estándar IEEE 754 de precisión simple [racionales]

- Existencia de casos especiales:

$(-1)^s * \mathbf{0}.\text{mantisa} * 2^{-126}$

Exponente	Mantisa	Valor especial
0 (0000 0000)	0	+/- 0 (según signo)
0 (0000 0000)	No cero	Número NO normalizado
255 (1111 1111)	No cero	NaN (0/0,…)
255 (1111 1111)	0	+/-infinito (según signo)
1-254	Cualquiera	Número normalizado (no especial)

$(-1)^s * \mathbf{1}.\text{mantisa} * 2^{\text{exponente}-127}$

Félix García-Carballeira, Alejandro Calderón Mateos

Ejemplos (incluyen casos especiales)

S	E	M	N
1	00000000	00000000000000000000000	-0 (Excepción 0) E=0 y M=0
1	01111111	00000000000000000000000	$-2^0 \times 1.0_2 = -1$
0	10000001	11100000000000000000000	$+2^2 \times 1.111_2 = +2^2 \times (2^0+2^{-1}+2^{-2}+2^{-3}) = +7.5$
0	11111111	00000000000000000000000	∞ (Excepción ∞) E=255 y M=0
0	11111111	10000000000000000000001	NaN (*Not a Number*) E=255 y M$\neq$0

Ejercicio

a) Calcular el valor correspondiente al número
0 10000011 11000000000000000000000
dado en coma flotante según norma 754 de simple precisión

Ejercicio (solución)

a) Calcular el valor correspondiente al número
0 10000011 11000000000000000000000
dado en coma flotante según norma 754 de simple precisión

a) Bit de signo: 0 $\Rightarrow (-1)^0 = +1$
b) Exponente: $10000011_2 = 131_{10} \Rightarrow E - 127 = 131 - 127 = 4$
c) Mantisa: 11000000000000000000000 $\Rightarrow 1 \times 2^{-1} + 1 \times 2^{-2} = 0{,}75$

Por tanto, el valor decimal del nº es $+1 \times 2^4 \times 1{,}75 = +28$

Félix García-Carballeira, Alejandro Calderón Mateos

Ejercicio

b) Expresar según norma IEEE 754 de simple precisión el nº -9

Ejercicio (solución)

b) Expresar según norma IEEE 754 de simple precisión el nº -9

$-9_{10} = -1001_2 = -1001_2 \times 2^0 = -1{,}001_2 \times 2^3$ (mantisa normalizada)

a) Bit de signo: negativo ⇨ S=1
b) Exponente: 3+127 (exceso) = 130 ⇨ 10000010
c) Mantisa: 1,001 (bit impl.) ⇨ 00100000000000000000000

Por tanto -9 = 1 10000010 00100000000000000000000

Félix García-Carballeira, Alejandro Calderón Mateos

Estándar IEEE 754 de precisión simple [racionales]

- Rango de magnitudes representables (sin considerar el signo):
  - Menor normalizado:
    $2^{1-127} \times 1.00000000000000000000000_2$
  - Mayor normalizado:
    $2^{254-127} \times 1.11111111111111111111111_2$
  - Menor no normalizado:
    $2^{-126} \times 0.00000000000000000000001_2$
  - Mayor no normalizado:
    $2^{-126} \times 0.11111111111111111111111_2$

$(-1)^s * 0.\text{mantisa} * 2^{-126}$

Exponente	Mantisa	Valor especial
0	**$\neq 0$**	**No normalizado**
1-254	**cualquiera**	**normalizado**

$(-1)^s * 1.\text{mantisa} * 2^{\text{exponente}-127}$

Félix García-Carballeira, Alejandro Calderón Mateos

Estándar IEEE 754 de precisión simple [racionales]

- Rango de magnitudes representables (sin considerar el signo):
  - Menor normalizado:
    $2^{1-127} \times 1.00000000000000000000000_2 = 2^{-126}$
  - Mayor normalizado:
    $2^{254-127} \times 1.11111111111111111111111_2 = 2^{127} \times (2 - 2^{-23}) = 2^{128} \times (1 - 2^{-24})$
  - Menor no normalizado:
    $2^{-126} \times 0.00000000000000000000001_2 = 2^{-149}$
  - Mayor no normalizado:
    $2^{-126} \times 0.11111111111111111111111_2 = 2^{-126} \times (1 - 2^{-23})$

Truco:

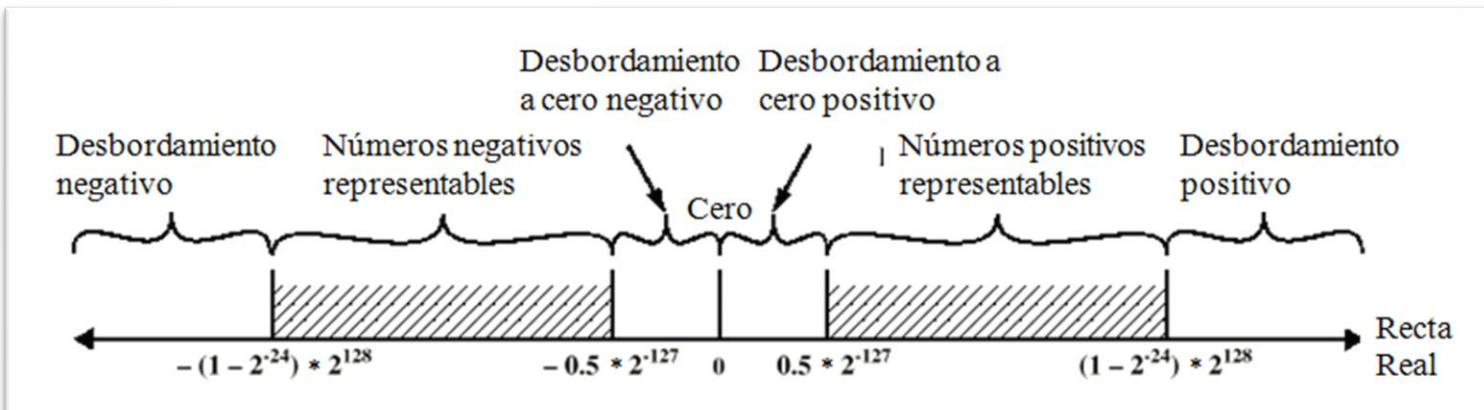
$$
\begin{array}{rrl}
 & 1.11111111111111111111111_2 & = X \\
+ & 0.00000000000000000000001_2 & = 2^{-23} \\
\hline
 & 10.00000000000000000000000_2 & = 2
\end{array}
$$

$$X = 2 - 2^{-23}$$

Félix García-Carballeira, Alejandro Calderón Mateos

Estándar IEEE 754 de precisión simple [racionales]

- Rango de magnitudes representables (sin considerar el signo):
  - Menor normalizado:
    $2^{1-127} \times 1.00000000000000000000000_2 = 2^{-126} = 2^{-127} \times 0.5$
  - Mayor normalizado:
    $2^{254-127} \times 1.11111111111111111111111_2 = 2^{127} \times (2 - 2^{-23}) = 2^{128} \times (1 - 2^{-24})$
  - Menor no normalizado:
    $2^{-126} \times 0.00000000000000000000001_2 = 2^{-149}$
  - Mayor no normalizado:
    $2^{-126} \times 0.11111111111111111111111_2 = 2^{-126} \times (1 - 2^{-23})$



Ejercicio

- ¿Cuántos números de *floats* (coma flotante de simple precisión) hay entre el 1 y el 2 (no incluido)?

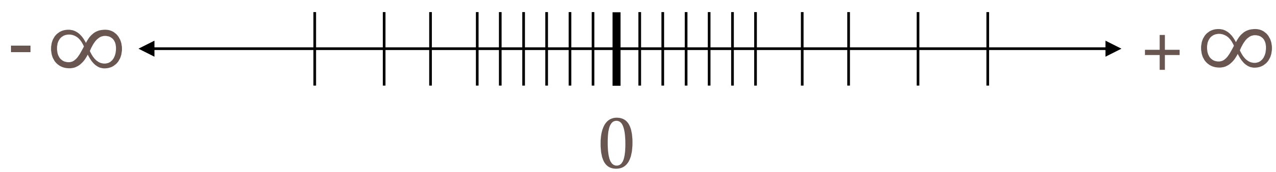
- ¿Cuántos números de *floats* (coma flotante de simple precisión) hay entre el 2 y el 3 (no incluido)?

Ejercicio (solución)

- ¿Cuántos números de *floats* (coma flotante de simple precisión) hay entre el 1 y el 2 (no incluido)?
  - $1 = 1{,}00000000000000000000000 \times 2^0$
  - $2 = 1{,}00000000000000000000000 \times 2^1$
  - Entre 1 y 2 hay 2^{23} números
- ¿Cuántos números de *floats* (coma flotante de simple precisión) hay entre el 2 y el 3 (no incluido)?
  - $2 = 1{,}00000000000000000000000 \times 2^1$
  - $3 = 1{,}10000000000000000000000 \times 2^1$
  - Entre 2 y 3 hay 2^{22} números

Números representables

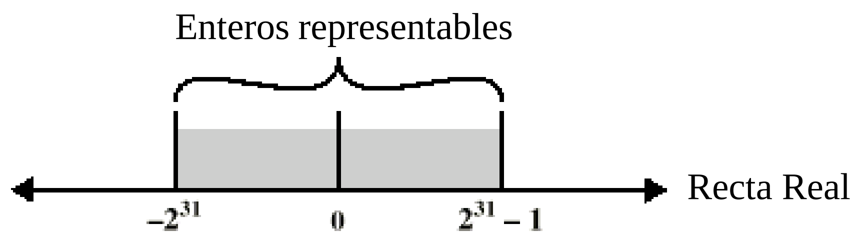
- Resolución variable:
  Más denso cerca de cero, menos hacia el infinito



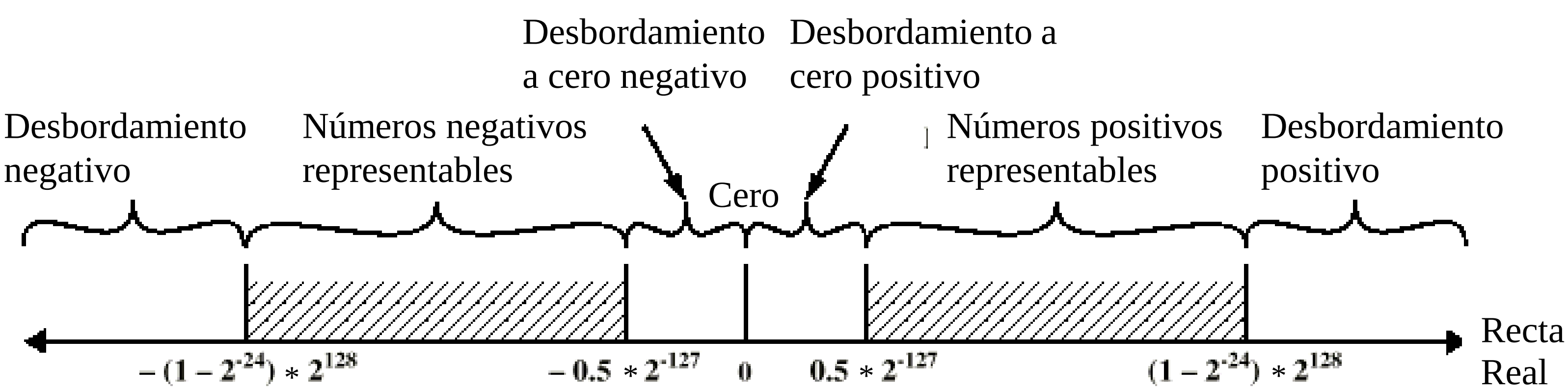
https://www.hobbyconsolas.com/reportajes/toy-story-espacio-disney-infinity-58154

Félix García-Carballeira, Alejandro Calderón Mateos

Números representables



(a) Enteros en complemento a dos



(b) Números en coma flotante

Ejemplo 1
imprecisión

0,4 →

0	01111101	10011001100110011001101

3.9999998×10^{-1}

0,1 →

0	01111011	10011001100110011001100

9.9999994×10^{-2}

Ejemplo 2
imprecisión

- ¿Cómo realiza C una división?

t2.c

```
#include <stdio.h>

int main ( )
{
   float a ;

   a = 3.0/7.0 ;
   if (a == 3.0/7.0)
          printf("Igual\n") ;
   else printf("No Igual\n") ;
   return (0) ;
}
```

Ejemplo 2
imprecisión

- ¿Cómo realiza C una división?

t2.c

```c
#include <stdio.h>

int main ( )
{
   float a ;

   a = 3.0/7.0 ;
   if (a == 3.0/7.0)
          printf("Igual\n") ;
   else printf("No Igual\n") ;
   return (0) ;
}
```

```
$ gcc -o t2 t2.c
$ ./t2
No Igual
```

Ejemplo 2
imprecisión

- ¿Cómo realiza C una división?

t2.c

```
#include <stdio.h>

int main ( )
{
   float a ;

   a = 3.0/7.0 ;
   if (a == 3.0/7.0)
         printf("Igual\n") ;
   else printf("No Igual\n") ;
   return (0) ;
}
```

float

double

```
$ gcc -o t2 t2.c
$ ./t2
No Igual
```


Félix García-Carballeira, Alejandro Calderón Mateos

Ejemplo 3
imprecisión

- La propiedad asociativa no siempre se cumple

  ¿ a + (b + c) = (a + b) + c ?

t1.c

```
#include <stdio.h>

int main ( )
{
      float x, y, z ;

      x = 10e30;   y = -10e30;   z = 1;
      printf("(x+y)+z = %f\n",(x+y)+z) ;
      printf("x+(y+z) = %f\n",x+(y+z)) ;

      return (0) ;
}
```

Ejemplo 3
imprecisión

- La propiedad asociativa no siempre se cumple

  ¿ a + (b + c) = (a + b) + c ?

t1.c

```c
#include <stdio.h>

int main ( )
{
      float x, y, z ;

      x = 10e30;   y = -10e30;   z = 1;
      printf("(x+y)+z = %f\n",(x+y)+z) ;
      printf("x+(y+z) = %f\n",x+(y+z)) ;

      return (0) ;
}
```

```
$ gcc -o t1 t1.c
$ ./t1
(x+y)+z = 1.000000
x+(y+z) = 0.000000
```

Ejemplo

Conversión int → float → int

```
if (i == (int)((float) i)) {
      printf("true");
}
```

- No siempre es cierto
- Muchos valores enteros grandes no tienen una representación exacta en coma flotante
- ¿Qué ocurre con `double`?

Ejemplo

- El número 133000405 en binario es:
  - 111111011010110110011010101 (27 bits)
- $111111011010110110011010101 \times 2^{0}$
- Se normaliza
  - $1,11111011010110110011010101 \times 2^{26}$
  - S = 0 (positivo)
  - $e = 26 \rightarrow \quad E = 26 + 127 = 153$
  - M = 11111011010110110011010 (se pierden los 3 últimos bits)
- El número realmente almacenado es
  - $1,11111011010110110011010 \times 2^{26} =$
  - $111111011010110110011010 \times 2^{3} = 133000400$

Ejemplo
Conversión float → int → float

```
if (f == (float)((int) f)) {
    printf("true");
}
```

- No siempre es cierto
- Los números con decimales no tienen representación entera

Redondeo

- El redondeo elimina cifras menos significativas de un número para obtener un valor aproximado.

- Tipos de redondeo:
  - Redondeo hacia + ∞
    - Redondeo "hacia arriba": 2.001 → 3, -2.001 → -2
  - Redondeo hacia - ∞
    - Redondea "hacia abajo": 1.999 → 1, -1.999 → -2
  - Truncar
    - Descarta los últimos bits: 1.299 → 1.2
  - Redondeo al más cercano
    - 2.4 → 2, 2.6 → 3, -1.4 → -1

Redondeo

- El redondeo supone ir perdiendo precisión.

- El redondeo ocurre:

  - Al pasar a una representación con menos representables:
    - Ej.: Un valor de doble a simple precisión
    - Ej.: Un valor en coma flotante a entero

  - Al realizar operaciones aritméticas:
    - Ej.: Después de sumar dos números en coma flotante (al usar dígitos de guarda)

Dígitos de guarda

- Se utilizan dígitos de guarda para mejorar la precisión: internamente se usan dígitos adicionales para operar.

- Ejemplo: $2,65 \times 10^0 + 2.34 \times 10^2$

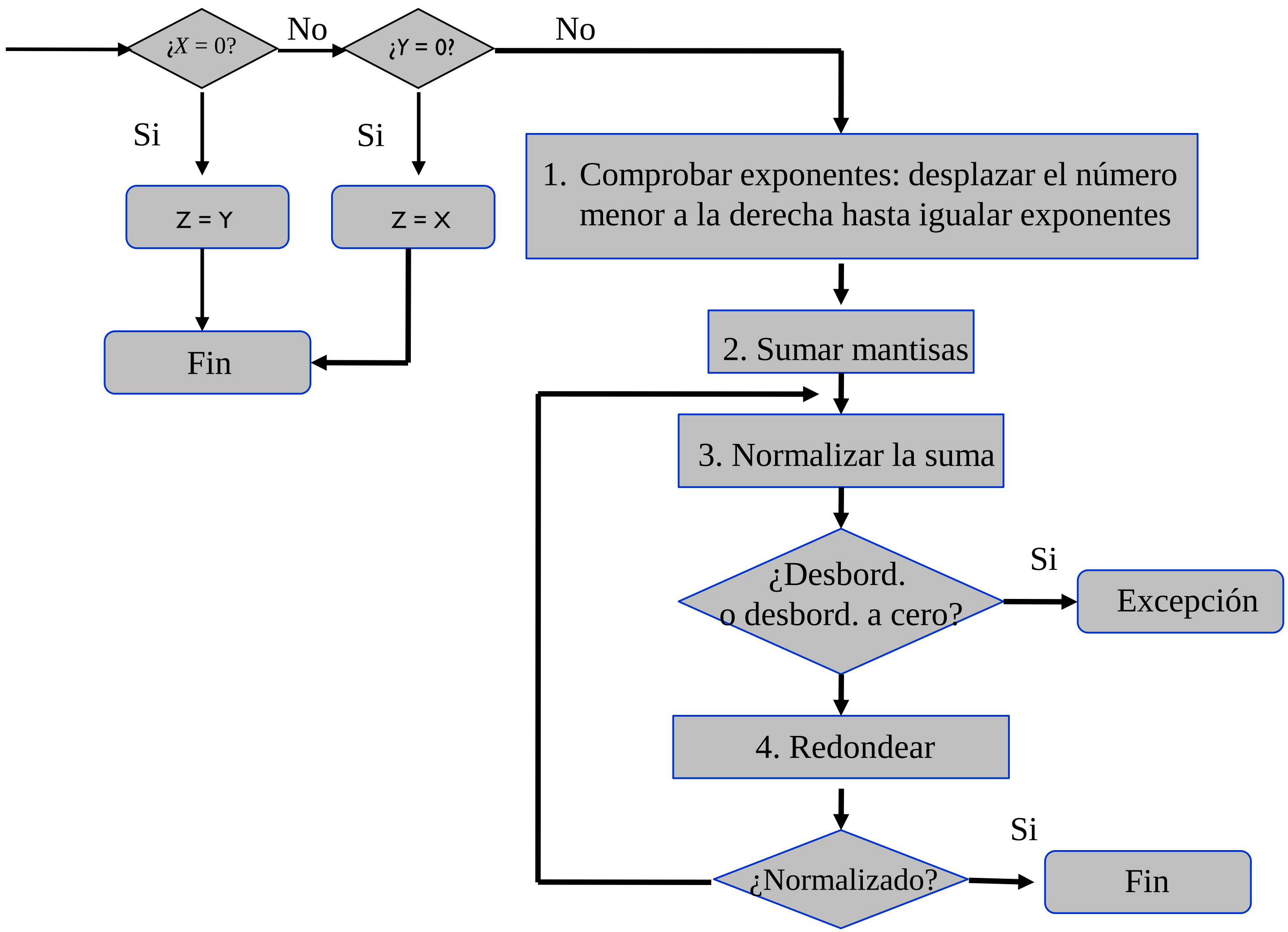
	SIN dígitos de guarda	CON dígitos de guarda
1.- igualar exponentes	$0,02 \times 10^2$ $+ 2,34 \times 10^2$	$0,0265 \times 10^2$ $+ 2,3400 \times 10^2$
2.- sumar	$2,36 \times 10^2$	$2,3665 \times 10^2$
3.- redondear	$2,36 \times 10^2$	$2,37 \times 10^2$

Operaciones en coma flotante

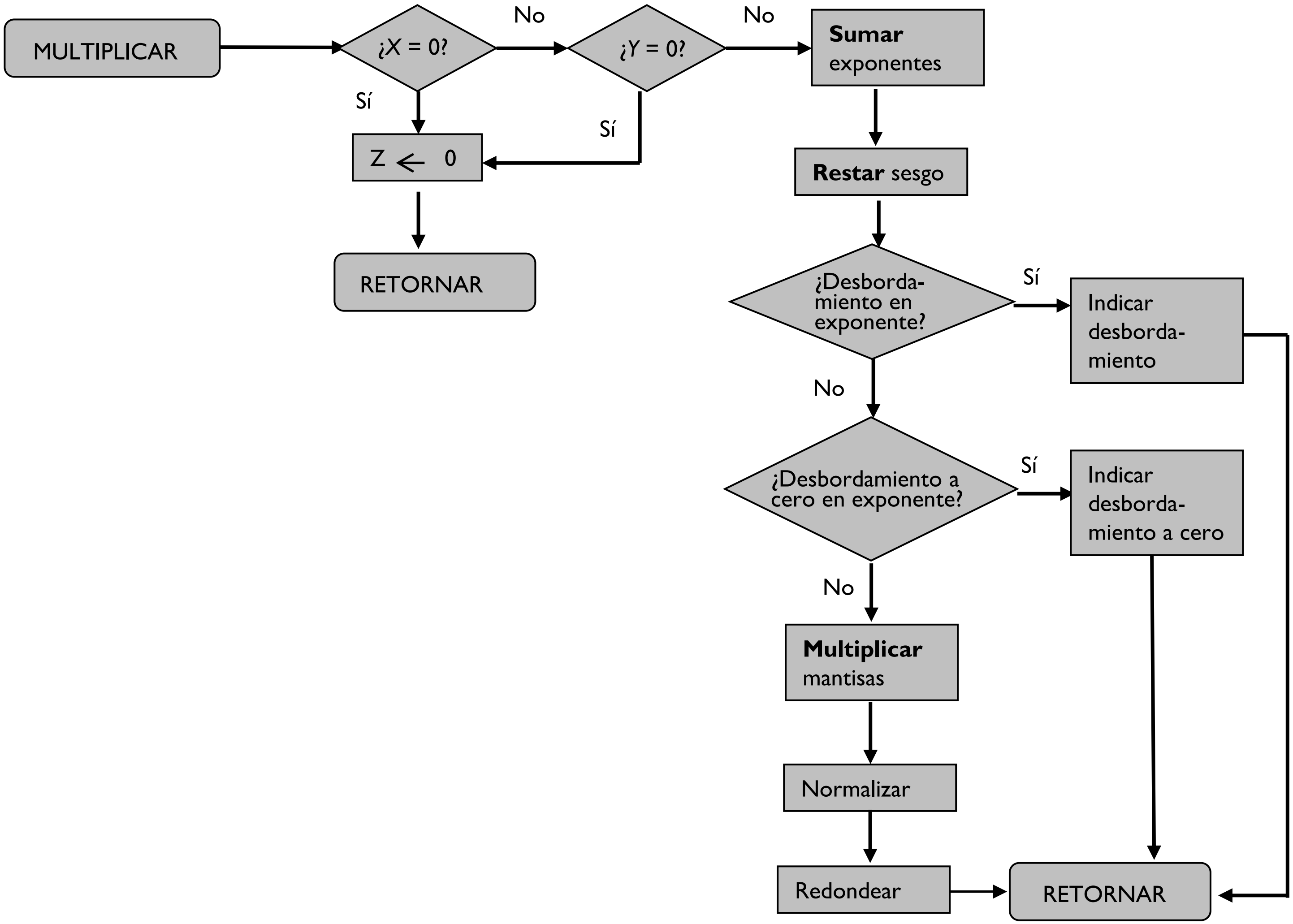
- Sumar
- Restar
  1. Comprobar valores cero.
  2. Igualar exponentes (desplazar número menor a la derecha).
  3. Sumar/restar las mantisas.
  4. Normalizar el resultado.

- Multiplicar
- Dividir
  1. Comprobar valores cero.
  2. Sumar/restar exponentes.
  3. Multiplicar/dividir mantisas (teniendo en cuenta el signo).
  4. Normalizar el resultado.
  5. Redondear el resultado.

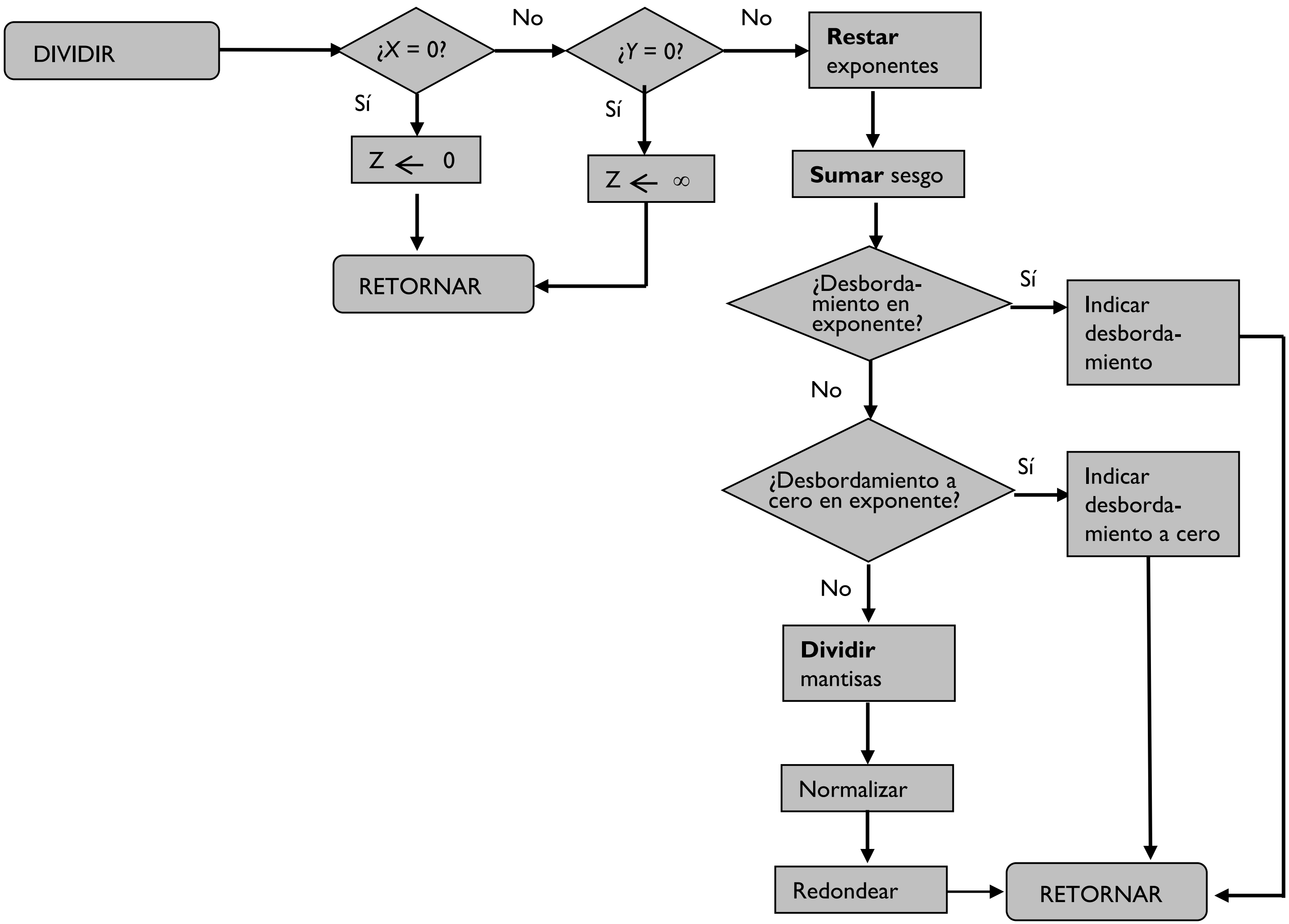
Suma y resta: Z=X+Y y Z=X-Y



Multiplicación: Z=X*Y



División: Z=X/Y



Félix García-Carballeira, Alejandro Calderón Mateos

Ejercicio

- Usando el formato IEEE 754, sumar 7,5 y 1,5 paso a paso

Solución

1) $7{,}5 + 1{,}5 =$
2) $1{,}111 * 2^2 + 1{,}1 * 2^0 =$
3) $1{,}111 * 2^2 + 0{,}011 * 2^2 =$
4) $10{,}010 * 2^2 =$
5) $1{,}0010 * 2^3$

Pasar a binario

Igualar exponentes

Sumar

Ajustar exponentes

Solución

- Representación de los números

7,5 → 0 10000001 11100000000000000000000

\+ **1,5** → 0 01111111 10000000000000000000000

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Se separa exponentes y mantisas y se añade el bit implícito

	7,5 →	0	10000001	1.11100000000000000000000
+	1,5 →	0	01111111	1.10000000000000000000000

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Igualar exponentes

7,5 →	0	10000001	1.11100000000000000000000
+ **1,5** →	0	01111111	1.10000000000000000000000

Solución

- Igualar exponentes

7,5 →	0	10000001	1.11100000000000000000000
		+1	1
+ **1,5** →	0	01111111	1.10000000000000000000000

Solución

- Igualar exponentes

7,5 →	0	10000001	1.11100000000000000000000
+ 1,5 →	0	10000000	0.11000000000000000000000

+1 1

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Igualar exponentes

	Signo	Exponente	Mantisa
7,5 →	0	10000001	1.11100000000000000000000
		+1	1
+ 1,5 →	0	10000001	0.01100000000000000000000

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Sumar mantisas

7,5 →	0	10000001	1.11100000000000000000000
+ **1,5** →	0	10000001	0.01100000000000000000000

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Normalizar el resultado

7,5 →	0	10000001	1.11100000000000000000000
+ **1,5** →	0	10000001	0.01100000000000000000000
9 →	0	10000001	10.01000000000000000000000

Se produce un acarreo, mantisa no normalizada

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Normalizar el resultado

7,5 →	0	10000001	1.11100000000000000000000
+ 1,5 →	0	10000001	0.01100000000000000000000

+1 1

9 →	0	10000001	10.01000000000000000000000

Solución

7,5 →	0	10000001	1.11100000000000000000000
+ **1,5** →	0	10000001	0.01100000000000000000000
9 →	0	10000010	1.00100000000000000000000

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Se almacena el resultado eliminando el bit implícito

9 → 0 10000010 00100000000000000000000

Ejercicio

- Usando el formato IEEE 754, multiplicar 7,5 y 1,5 paso a paso

Solución

- Representación de los números

9 → 0 10000010 00100000000000000000000

\- 7,5 → 1 10000001 11100000000000000000000

Solución

- Se separan exponentes y mantisas y se añade bit implícito

	7,5 →	0	10000001	1.11100000000000000000000
X	**1,5** →	0	01111111	1.10000000000000000000000

Se añade el bit implícito para operar

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Multiplicar: sumar exponentes y multiplicar mantisas

	7,5 →	0	10000001	1.11100000000000000000000
X	**1,5** →	0	01111111	1.10000000000000000000000
			+	×
		0	100000000	10.1101000000000000000000

Solución

- Multiplicar: quitar el sesgo al exponente (hay dos)

7,5 →

0	10000001	1.11100000000000000000000

X 1,5 →

0	01111111	1.10000000000000000000000

0	100000000	10.11010000000000000000000

\- 01111111

Solución

- Multiplicar: normalizar el resultado

7,5 →	0	10000001	1.11100000000000000000000
X 1,5 →	0	01111111	1.10000000000000000000000
		+1	1
11,25	0	10000001	10.11010000000000000000000

Solución

- Resultado normalizado…

7,5 →	0	10000001	1.11100000000000000000000
X 1,5 →	0	01111111	1.10000000000000000000000
11,25	0	10000010	1.011010000000000000000000

Félix García-Carballeira, Alejandro Calderón Mateos

Solución

- Se almacena el resultado eliminando el bit implícito

11,25 0 10000010 011010000000000000000000

Félix García-Carballeira, Alejandro Calderón Mateos

Evolución de IEEE 754

- 1985 – IEEE 754
- 2008 – IEEE 754-2008 (754+854)
- 2011 – ISO/IEC/IEEE 60559:2011 (754-2008)
- 2019 – IEEE 754-2019 (aclaraciones y arreglos)

Name	Common name	Base	Significand bits/digits	Exp. bits	Decimal E max	Exponent bias[14]	E min	E max
binary16	Half precision	2	11	5	4.51	24−1 = 15	−14	+15
binary32	Single precision	2	24	8	38.23	27−1 = 127	−126	+127
binary64	Double precision	2	53	11	307.95	210−1 = 1023	−1022	+1023
binary128	Quadruple precision	2	113	15	4931.77	214−1 = 16383	−16382	+16383
binary256	Octuple precision	2	237	19	78913.2	218−1 = 262143	−262142	+262143
decimal32		10	7	7.58	96	101	−95	+96
decimal64		10	16	9.58	384	398	−383	+384
decimal128		10	34	13.58	6144	6176	−6143	+6144

Grupo ARCOS

uc3m | Universidad Carlos III de Madrid

Tema 2: Representación de la información

Estructura de Computadores

Grado en Ingeniería Informática

Grado en Matemática aplicada y Computación

Doble Grado en Ingeniería Informática y Administración de Empresas