

Tema 2

Representación de Información

atimizamos un alfabeto, o un conjunto de símbolos. Cada símbolo es denominado un **character**.

Cada character tiene algun significado: sonidos, pausas, numeros, sentimientos.

Caracteres alfabéticos

Representan las **26 letras** del alfabeto inglés. Desde "a" hasta "z", en mayúsculas o minúsculas.

Caracteres alfabéticos

Representan las **26 letras** del alfabeto inglés. Desde "a" hasta "z", en mayúsculas o minúsculas.

Caracteres alfabéticos

Representan las **26 letras** del alfabeto inglés. Desde "a" hasta "z", en mayúsculas o minúsculas.

- abcdefghijklmnopqrstuvwxyz

Caracteres alfabéticos

Representan las **26 letras** del alfabeto inglés. Desde "a" hasta "z", en mayúsculas o minúsculas.

- abcdefghijklmnopqrstuvwxyz
- ABCDEFGHIJKLMNOPQRSTUVWXYZ

Caracteres Numéricos

Representan los números de base decimal. Desde el "0" hasta el "9".

Caracteres Numéricos

Representan los números de base decimal. Desde el "0" hasta el "9".

Caracteres Numéricos

Representan los números de base decimal. Desde el "0" hasta el "9".

- 0123456789

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

- SPACE

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

- SPACE
- () {} [] < >

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

- SPACE
- () {} [] < >
- @

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

- SPACE
- () {} [] < >
- @
- ¡ ! ¿ ?

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

- SPACE
- () {} [] < >
- @
- ¡ ! ¿ ?
- , . ;

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

- SPACE
- () {} [] < >
- @
- ¡ ! ¿ ?
- , . ;
- = + - / * %

Caracteres especiales

Estos caracteres forman parte de alfabetos de otros lenguajes como el español, francés, alemán, etc. También incluye símbolos que indican énfasis, preguntas, enumeraciones, etc.

- SPACE
- () {} [] < >
- @
- ¡ ! ¿ ?
- , . ;
- = + - / * %
- Ñ ß á ë î

Caracteres de Control

Son caracteres que no tienen representación gráfica pero sirven de control para indicar espacios tabulados, segmentos, saltos de línea, etc.

Caracteres de Control

Son caracteres que no tienen representación gráfica pero sirven de control para indicar espacios tabulados, segmentos, saltos de línea, etc.

Caracteres de Control

Son caracteres que no tienen representación gráfica pero sirven de control para indicar espacios tabulados, segmentos, saltos de línea, etc.

- TAB

Caracteres de Control

Son caracteres que no tienen representación gráfica pero sirven de control para indicar espacios tabulados, segmentos, saltos de línea, etc.

- TAB
- NULL

Caracteres de Control

Son caracteres que no tienen representación gráfica pero sirven de control para indicar espacios tabulados, segmentos, saltos de línea, etc.

- TAB
- NULL
- BELL

Caracteres de Control

Son caracteres que no tienen representación gráfica pero sirven de control para indicar espacios tabulados, segmentos, saltos de línea, etc.

- TAB
- NULL
- BELL
- Backspace

Caracteres de Control

Son caracteres que no tienen representación gráfica pero sirven de control para indicar espacios tabulados, segmentos, saltos de línea, etc.

- TAB
- NULL
- BELL
- Backspace
- Line Feed

Caracteres de Control

Son caracteres que no tienen representación gráfica pero sirven de control para indicar espacios tabulados, segmentos, saltos de línea, etc.

- TAB
- NULL
- BELL
- Backspace
- Line Feed
- Carriage Return

Caracteres gráficos

Sirven para representar caracteres de alfabetos chinos, japoneses, árabes, emojis, e íconos gráficos. También podemos incluir **emojis** en esta categoría.

Caracteres gráficos

Sirven para representar caracteres de alfabetos chinos, japoneses, árabes, emojis, e íconos gráficos. También podemos incluir **emojis** en esta categoría.

Caracteres gráficos

Sirven para representar caracteres de alfabetos chinos, japoneses, árabes, emojis, e íconos gráficos. También podemos incluir **emojis** en esta categoría.

- © ☒ ® ☒

Caracteres gráficos

Sirven para representar caracteres de alfabetos chinos, japoneses, árabes, emojis, e íconos gráficos. También podemos incluir **emojis** en esta categoría.

- ® ☒ ® ☒
- → 殭 ☒ ☒ ☒ ☒ ★

Caracteres gráficos

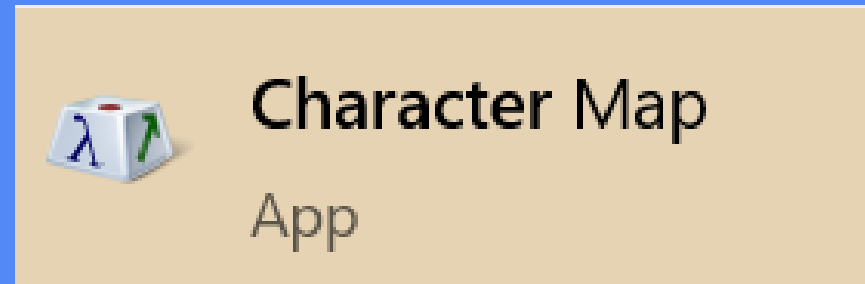
Sirven para representar caracteres de alfabetos chinos, japoneses, árabes, emojis, e íconos gráficos. También podemos incluir **emojis** en esta categoría.

- ® ☒ ® ☒
- → 殭 ☒ ☒ ☒ ☒ ☆
- 🙄 ☀️ 🚗 🍲 😂

En Windows, puedes acceder a la lista de las iconas disponibles con el programa "Character Map".

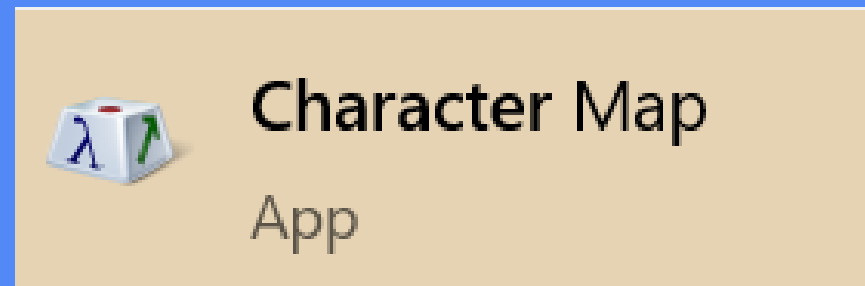
Character Map

Character Map

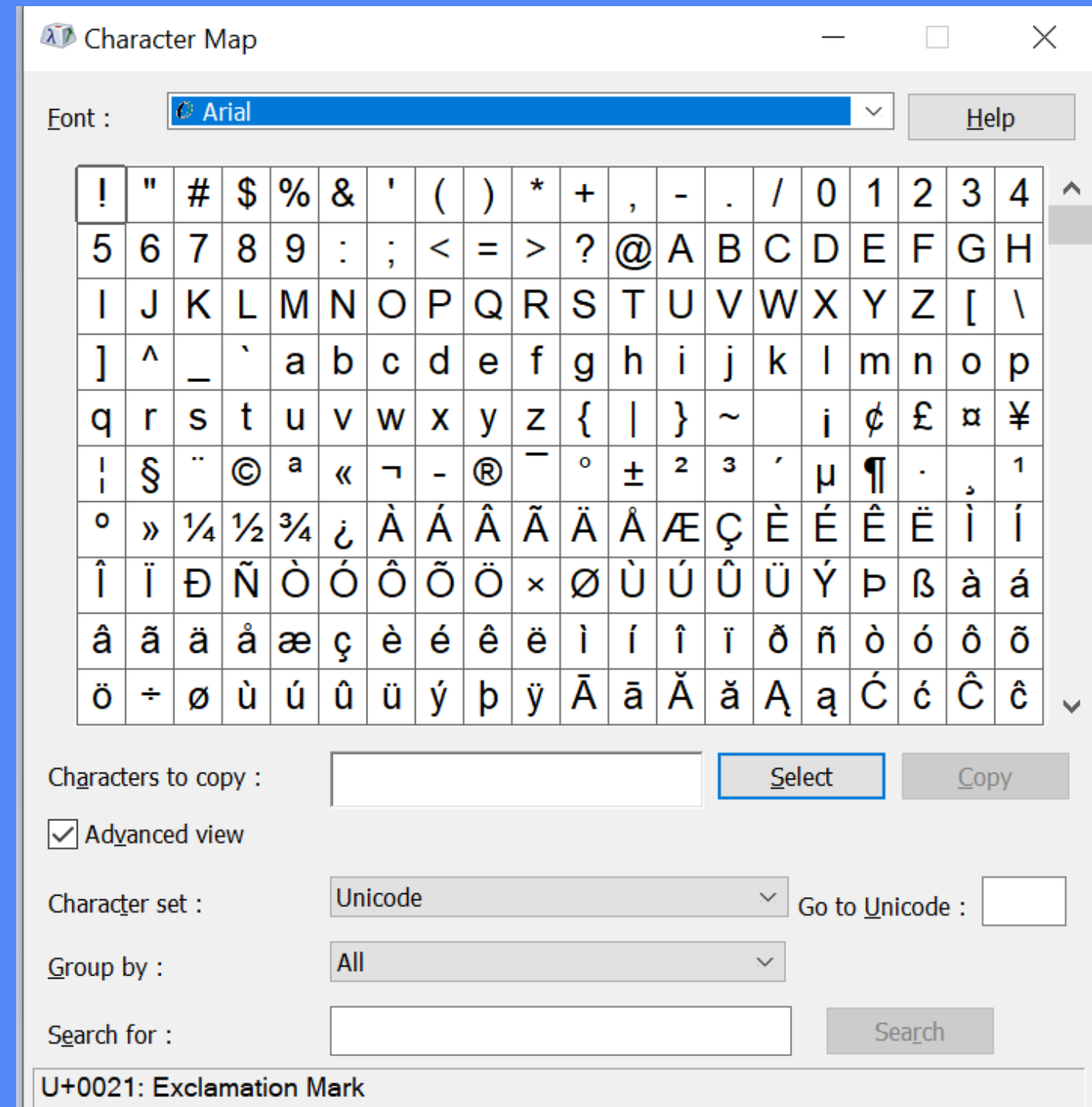


En Windows, puedes acceder a la lista de los íconos disponibles con el programa "Character Map".

Character Map



En Windows, puedes acceder a la lista de los íconos disponibles con el programa "Character Map".



Todas las instrucciones o interacciones con computadores se pueden **interpretar** a través de caracteres.

Los caracteres, internamente, están formados por **bits**.

Bits

Binary DigIT

Un bit es la unidad más básica de información en el mundo digital. Un bit puede representar dos valores: 0 ó 1.

Estos valores se pueden interpretar también como **TRUE** o **FALSE**, o los estados **encendido** o **apagado**.

Para una circuito es fácil representarlo: +3.5 volts ó 0 volts.





¿Por qué utilizamos bits?

¿Por qué utilizamos bits?

Es más fácil trabajar con 2 rangos de valores que con 3,4,5[...].

¿Por qué utilizamos bits?

Es más fácil trabajar con 2 rangos de valores que con 3,4,5[...].

Los transistores manejan dos estados: encendidos y apagados.

¿Por qué utilizamos bits?

Es más fácil trabajar con 2 rangos de valores que con 3,4,5[...].

Los transistores manejan dos estados: encendidos y apagados.

El almacenamiento y procesamiento en binario es muy confiable.

Transi



Video Playback Disabled

Almacenamiento de información

Ejemplo:

- De 0 hasta 1.75 = 0

Almacenamiento de información

Ejemplo:

- De 0 hasta 1.75 = 0
- De 1.76 hasta 3.5 = 1

1.76	2	1	0
2.5	2.2	1.2	0.9
0.8	1.6	1.74	2.2

Almacenamiento de información

Ejemplo:

- De 0 hasta 1.75 = 0
- De 1.76 hasta 3.5 = 1

Almacenamiento de información

Ejemplo:

- De 0 hasta 1.75 = 0
- De 1.76 hasta 3.5 = 1

1.76	2	1	0
2.5	2.2	1.2	0.9
0.8	1.6	1.74	2.2
	1	1	0 0

Almacenamiento de información

Ejemplo:

- De 0 hasta 1.75 = 0
- De 1.76 hasta 3.5 = 1

1.76	2	1	0
2.5	2.2	1.2	0.9
0.8	1.6	1.74	2.2
1	1	0	0
1	1	0	0

Almacenamiento de información

Ejemplo:

- De 0 hasta 1.75 = 0
- De 1.76 hasta 3.5 = 1

1.76	2	1	0
2.5	2.2	1.2	0.9
0.8	1.6	1.74	2.2

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

Bytes y Bits

Cuando la computadora requiere representar información mayor a un sólo bit, puede utilizar un **byte**.

8 bits juntos conforman 1 byte.

En ciertas codificaciones, **1 byte es suficiente para representar un caracter.**

Información

De acuerdo con la **JEDEC Solid State Technology Association**:

1 Kilobyte = 2^{10} bytes = 1024 bytes

1 Megabyte = 2^{20} bytes = 1024 Kilobytes

1 Gigabyte = 2^{30} bytes = 1024 Megabytes

1 Terabyte = 2^{40} bytes = 1024 Gigabytes

1 Pentabyte = 2^{50} bytes = 1024 Terabytes

Dato curioso

Hay una lucha entre el **material científico y la publicidad**, pues los prefijos Kilo, Mega, Giga, etc. indican potencias de 10, y no potencias de dos.

Sin embargo, al utilizar el Byte como la medida base de tamaño de almacenamiento hace que **utilizar potencias de 2 sea más común**.



Tamaño de discos

10.5 "Leopard"		10.6 "Snow Leopard"	
Name	Size	Name	Size
▼ The Very Best of Elvis Costello	294.7 MB	▼ The Very Best of Elvis Costello	309 MB
1-01 (What's So Fu... Understanding.m4a	6.7 MB	1-01 (What's So F...nderstanding.m4a	7 MB
1-02 Oliver's Army.m4a	5.7 MB	1-02 Oliver's Army.m4a	6 MB
1-03 Watching the Detectives.m4a	7.1 MB	1-03 Watching the Detectives.m4a	7.4 MB
1-04 Alison.m4a	6.4 MB	1-04 Alison.m4a	6.7 MB
1-05 (I Don't Want to Go to) Chelsea.m4a	6.2 MB	1-05 (I Don't Wan...o to) Chelsea.m4a	6.5 MB
1-06 Accidents Will Happen.m4a	5.7 MB	1-06 Accidents Will Happen.m4a	6 MB
1-07 Pump It Up.m4a	6.1 MB	1-07 Pump It Up.m4a	6.4 MB
1-08 I Can't Stand ...or Falling Down.m4a	4.1 MB	1-08 I Can't Stan... Falling Down.m4a	4.3 MB
1-09 Radio Radio.m4a	5.9 MB	1-09 Radio Radio.m4a	6.2 MB

Tamaño de discos

10.5 "Leopard"		10.6 "Snow Leopard"	
Name	Size	Name	Size
▼ The Very Best of Elvis Costello	294.7 MB	▼ The Very Best of Elvis Costello	309 MB
1-01 (What's So Fu... Understanding.m4a	6.7 MB	1-01 (What's So F...nderstanding.m4a	7 MB
1-02 Oliver's Army.m4a	5.7 MB	1-02 Oliver's Army.m4a	6 MB
1-03 Watching the Detectives.m4a	7.1 MB	1-03 Watching the Detectives.m4a	7.4 MB
1-04 Allison.m4a	6.4 MB	1-04 Allison.m4a	6.7 MB
1-05 (I Don't Want to Go to) Chelsea.m4a	6.2 MB	1-05 (I Don't Wan...o to) Chelsea.m4a	6.5 MB
1-06 Accidents Will Happen.m4a	5.7 MB	1-06 Accidents Will Happen.m4a	6 MB
1-07 Pump It Up.m4a	6.1 MB	1-07 Pump It Up.m4a	6.4 MB
1-08 I Can't Stand ...or Falling Down.m4a	4.1 MB	1-08 I Can't Stan... Falling Down.m4a	4.3 MB
1-09 Radio Radio.m4a	5.9 MB	1-09 Radio Radio.m4a	6.2 MB

Cuando Snow Leopard salió al mercado, Apple modificó la forma en la que Mac OS X calcula los tamaños de los archivos.

Mac OS X comenzó a utilizar potencias de 10 (1 KB = 1000 bytes)

Windows continúa utilizando potencias de 2 (1 KB = 1024 bytes)

Size on disk: 0.00 KB (0,192 bytes)

Created: Sunday, September 1, 2019, 12:06:04 PM

Modified: Sunday, September 1, 2019, 12:06:04 PM

Accessed: Today, September 3, 2019, 29 minutes ago

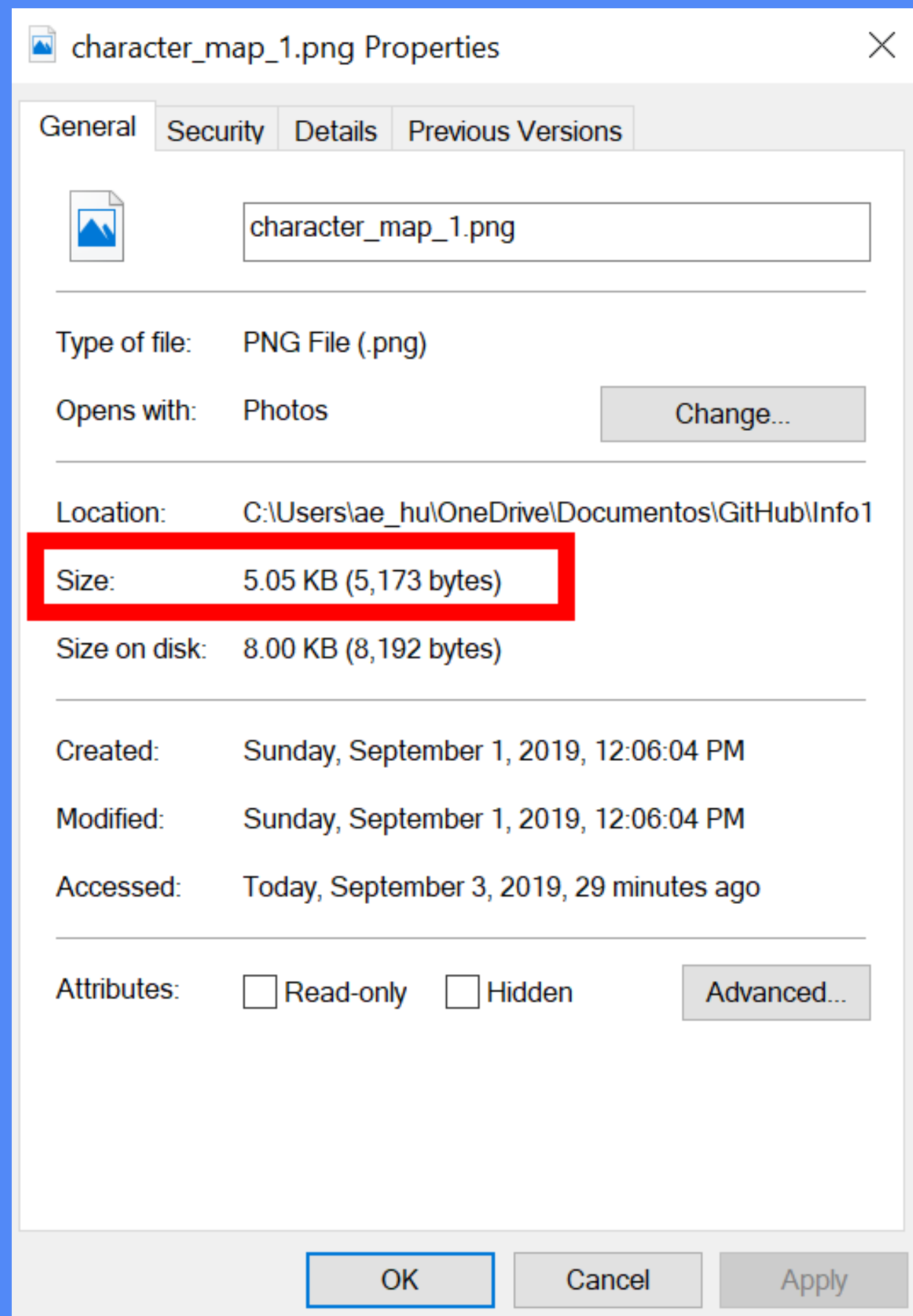
Attributes: ☐ Read-only ☐ Hidden

Advanced...

OK

Cancel

Apply



Si queremos conocer el tamaño exacto de un archivo, hay que abrir sus propiedades.

Sistema Decimal

Para representar números, las personas utilizamos el sistema decimal.

Esto significa que la base que usamos es 10, es decir, cada posición puede indicar solamente **10 valores distintos**.

Para representar números, las personas utilizamos el sistema decimal.

Esto significa que la base que usamos es 10, es decir, cada posición puede indicar solamente **10 valores distintos**.

NÚMERO MENOR
0

Para representar números, las personas utilizamos el sistema decimal.

Esto significa que la base que usamos es 10, es decir, cada posición puede indicar solamente **10 valores distintos**.

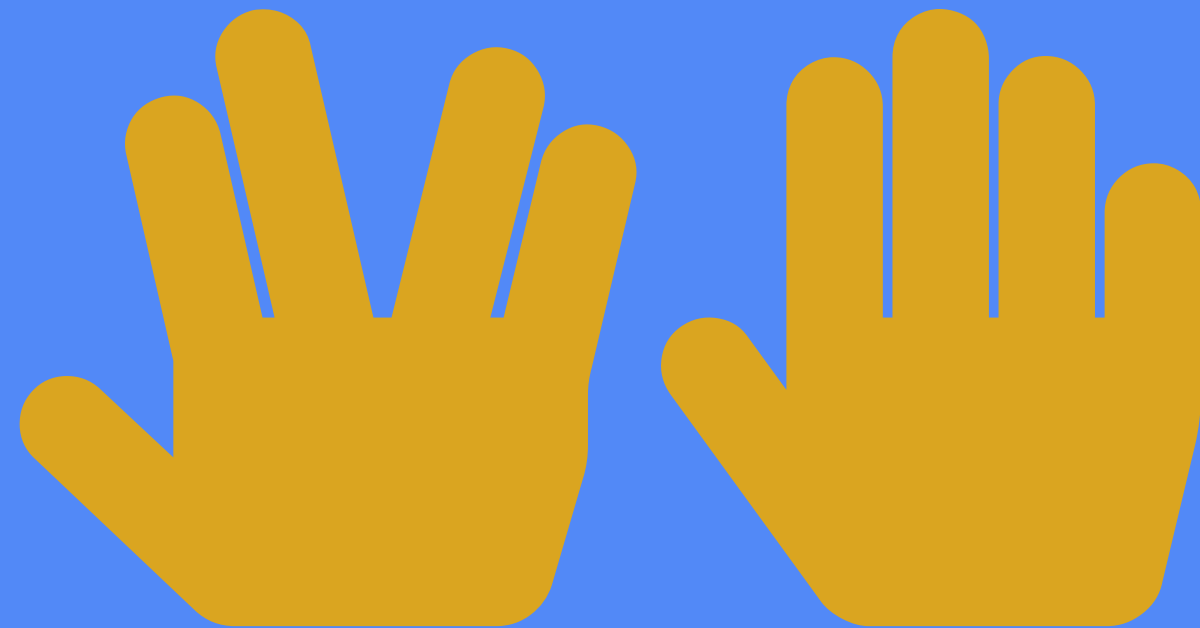


Las personas usamos este sistema para representar dinero, matrículas, numeraciones, etc.

Las personas usamos este sistema para representar dinero, matrículas, numeraciones, etc.



Las personas usamos este sistema para representar dinero, matrículas, numeraciones, etc.



Para indicar que estamos utilizando base 10,
agregamos el subíndice "10"

Ejemplo:

Ejemplo: $= 7_{10}$

Ejemplo: $= 257_{10}$

Ejemplo: $= 1000_{10}$

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21

Comenzamos con el número de menor valor

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

En cada salto, incrementamos el valor en 1

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

En cada salto, incrementamos el valor en 1

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

En cada salto, incrementamos el valor en 1

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Llegamos al último posible valor en la escala decimal.

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Agregamos un 1 al principio, y volvemos a comenzar desde 0

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Y así continuamos sucesivamente!

Sistema Binario

Binario

Las computadoras, representan la información utilizando el sistema binario. La base que utiliza es 2, es decir, cada posición puede indicar solamente **2 valores distintos**.

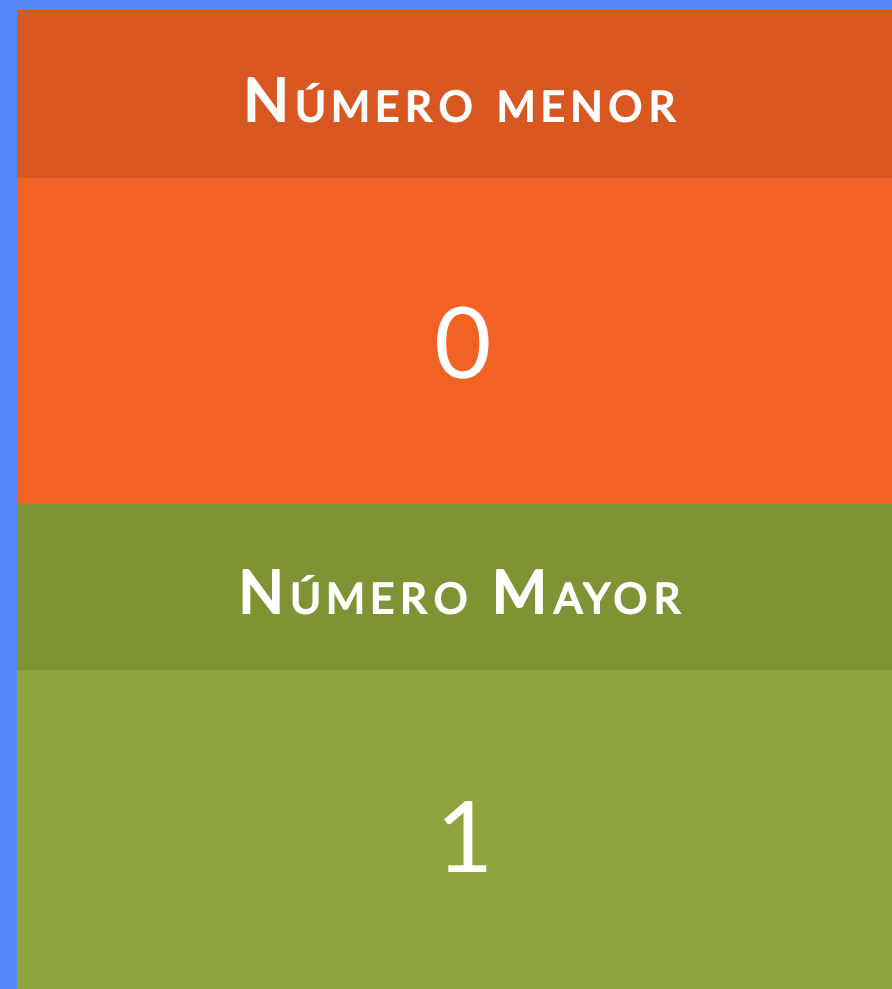
Binario

Las computadoras, representan la información utilizando el sistema binario. La base que utiliza es 2, es decir, cada posición puede indicar solamente **2 valores distintos**.

NÚMERO MENOR
0

Binario

Las computadoras, representan la información utilizando el sistema binario. La base que utiliza es 2, es decir, cada posición puede indicar solamente **2 valores distintos**.

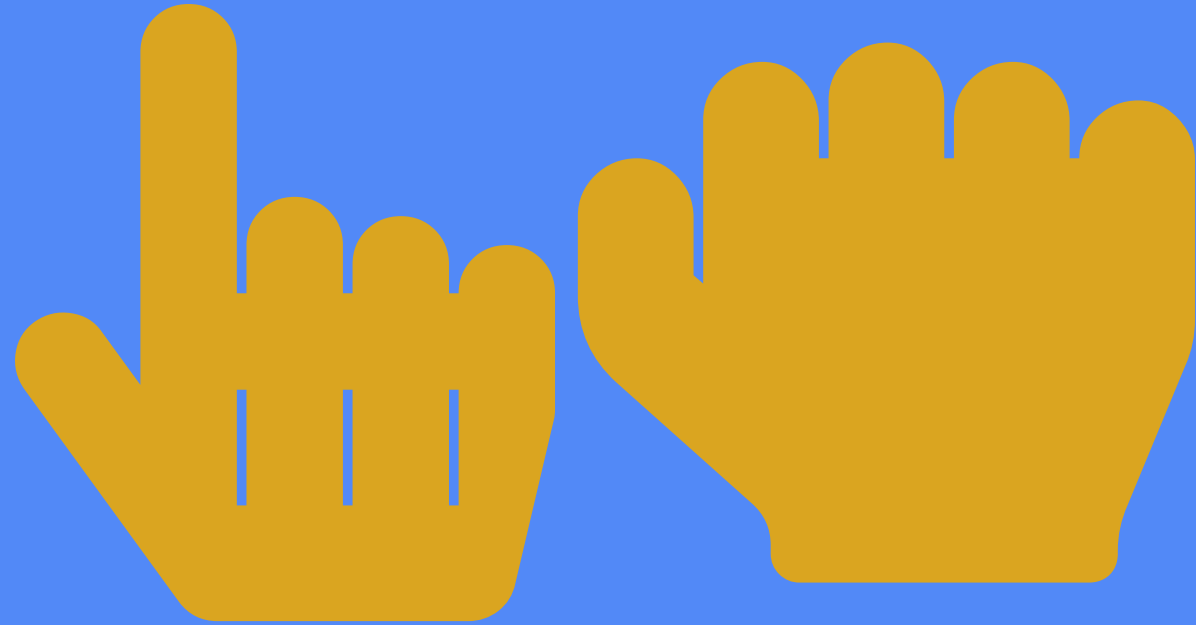


Las personas utilizan este sistema para representar
characters, contar, sumar, etc.

Las personas utilizan este sistema para representar
caracters, contar, sumar, etc.



Las personas utilizan este sistema para representar
caracters, contar, sumar, etc.



0

1

10

11

100

101

110

111

1000

1001

1010

1011

1100

1101

1110

1111

...

0

1

10

11

100

101

110

111

1000

1001

1010

1011

1100

1101

1110

1111

...

Comenzamos con el número de menor valor

0
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
...

Llegamos al último posible valor en la escala binaria

0
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
...

Agregamos un 1 al principio, y volvemos a comenzar desde 0

0
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
...

Sumamos 1, llegamos al máximo valor que se puede representar con dos dígitos

0
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
...

Agregamos un 1 al principio, y volvemos a comenzar desde 100

0
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
...

Agregamos un 1 al principio, y volvemos a comenzar desde 1000

0

1

10

11

100

101

110

111

1000

1001

1010

1011

1100

1101

1110

1111

...

0
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
...

Y así continuamos sucesivamente!

Conversiones de Binario a Decimal

1. Dado un número en binario:

110100_2

Conversiones de Binario a Decimal

1. Dado un número en binario:

110100_2

2. Comenzamos numerando las posiciones de derecha a izquierda, comenzando en cero.

$$110100_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

$$110100_2 = 32_{10} + 16_{10} + 0_{10} + 4_{10} + 0_{10} + 0_{10}$$

Conversiones de Binario a Decimal

1. Dado un número en binario:

$$110100_2$$

2. Comenzamos numerando las posiciones de derecha a izquierda, comenzando en cero.

$$110100_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

$$110100_2 = 32_{10} + 16_{10} + 0_{10} + 4_{10} + 0_{10} + 0_{10}$$

3. Completamos la suma:

$$110100_2 = 52_{10}$$

Otro ejemplo...

Problema 1:

$$10101011_2$$

$$10101011_2 = 1 \cdot 2^7 + \cancel{0 \cdot 2^6} + 1 \cdot 2^5 + \cancel{0 \cdot 2^4} + 1 \cdot 2^3 + \cancel{0 \cdot 2^2} + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$10101011_2 = 1 \cdot 2^7 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$10101011_2 = 128_{10} + 32_{10} + 8_{10} + 2_{10} + 1_{10}$$

$$10101011_2 = 171_{10}$$

1000_{10} :

Conversiones de Decimal a Binario

Dividimos el número a convertir
entre 2, que es la base deseada

$$1000_{10} : \\ 2 \overline{) 1000} \quad | \quad 0$$

Conversiones de Decimal a Binario

Apuntamos el valor del residuo.

$1000_{10} :$

$2 \overline{) 1000} \quad 0$

$2 \overline{) 500} \quad 0$

Conversiones de Decimal a Binario

Se baja el valor del resultado de la división. Se repite la operación y apuntamos el residuo.

$1000_{10} :$

$2 \overline{) 1000} \quad 0$

$2 \overline{) 500} \quad 0$

$2 \overline{) 250} \quad 0$

Conversiones de Decimal a Binario

Se baja el valor del resultado de la división. Se repite la operación y apuntamos el residuo.

$1000_{10} :$

$2 \overline{) 1000} \quad 0$

$2 \overline{) 500} \quad 0$

$2 \overline{) 250} \quad 0$

$2 \overline{) 125} \quad 1$

Conversiones de Decimal a Binario

El resultado de la división entera
 $125/2 = 62$, con un residuo de 1.

Apuntamos el residuo y el
resultado de la división entera.

$1000_{10} :$

$2 \overline{) 1000}$ 0

$2 \overline{) 500}$ 0

$2 \overline{) 250}$ 0

$2 \overline{) 125}$ 1

$2 \overline{) 62}$ 0

Conversiones de Decimal a Binario

Repetimos la operación hasta que
no podamos dividir el número.

$1000_{10} :$

$2 \overline{) 1000}$ 0

$2 \overline{) 500}$ 0

$2 \overline{) 250}$ 0

$2 \overline{) 125}$ 1

$2 \overline{) 62}$ 0

$2 \overline{) 31}$ 1

Conversiones de Decimal a Binario

Repetimos la operación hasta que
no podamos dividir el número.

$1000_{10} :$

$2 \overline{) 1000}$ 0

$2 \overline{) 500}$ 0

$2 \overline{) 250}$ 0

$2 \overline{) 125}$ 1

$2 \overline{) 62}$ 0

$2 \overline{) 31}$ 1

$2 \overline{) 15}$ 1

Conversiones de Decimal a Binario

Repetimos la operación hasta que
no podamos dividir el número.

$1000_{10} :$

$2 \overline{) 1000}$ 0

$2 \overline{) 500}$ 0

$2 \overline{) 250}$ 0

$2 \overline{) 125}$ 1

$2 \overline{) 62}$ 0

$2 \overline{) 31}$ 1

$2 \overline{) 15}$ 1

$2 \overline{) 7}$ 1

Conversiones de Decimal a Binario

Repetimos la operación hasta que
no podamos dividir el número.

$1000_{10} :$

$2 \overline{) 1000}$ 0

$2 \overline{) 500}$ 0

$2 \overline{) 250}$ 0

$2 \overline{) 125}$ 1

$2 \overline{) 62}$ 0

$2 \overline{) 31}$ 1

$2 \overline{) 15}$ 1

$2 \overline{) 7}$ 1

$2 \overline{) 3}$ 1

Conversiones de Decimal a Binario

Repetimos la operación hasta que
no podamos dividir el número.

$1000_{10} :$

$2 \overline{) 1000}$ 0

$2 \overline{) 500}$ 0

$2 \overline{) 250}$ 0

$2 \overline{) 125}$ 1

$2 \overline{) 62}$ 0

$2 \overline{) 31}$ 1

$2 \overline{) 15}$ 1

$2 \overline{) 7}$ 1

$2 \overline{) 3}$ 1

$2 \overline{) 1}$ 1

Conversiones de Decimal a Binario

La división entera $1/2 = 0$, con un residuo de 1. Apuntamos el 1 y concluimos.

Conversiones de Decimal a Binario

$$\begin{array}{r|l} 1000_{10} : & \\ 2 \overline{) 1000} & 0 \\ 2 \overline{) 500} & 0 \\ 2 \overline{) 250} & 0 \\ 2 \overline{) 125} & 1 \\ 2 \overline{) 62} & 0 \\ 2 \overline{) 31} & 1 \\ 2 \overline{) 15} & 1 \\ 2 \overline{) 7} & 1 \\ 2 \overline{) 3} & 1 \\ 2 \overline{) 1} & 1 \end{array} \Bigg\} = 1111101000_2$$

Para finalizar, copiamos los
números de abajo hacia arriba.

Trabajo individual

$$100001_2 \rightarrow x_{10}$$

$$1111111_2 \rightarrow x_{10}$$

$$254_{10} \rightarrow x_2$$

$$17_{10} \rightarrow x_2$$

Sistema Hexadecimal y Octal

Hexadecimal y Octal

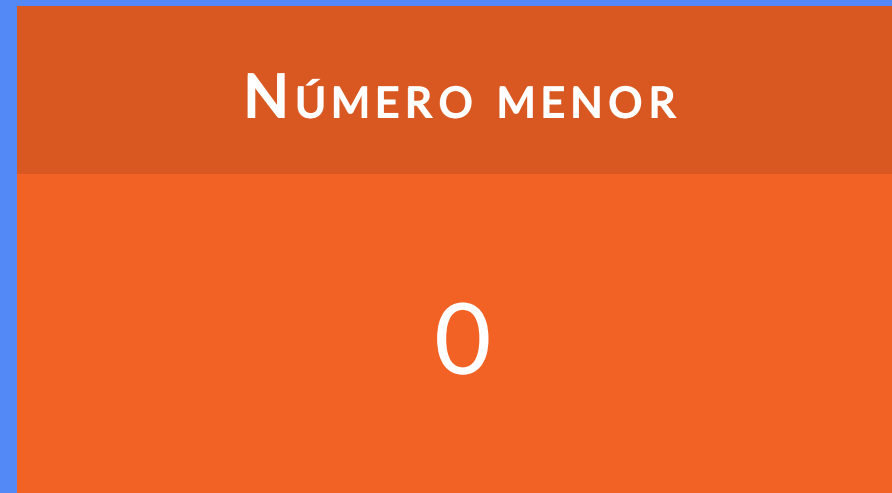
Para evitar largos conjuntos de caracteres cuando queramos representar números en binario, utilizamos las bases intermedias de **Octal** y **Hexadecimal**.

Octal

Cada dígito en esta representación puede representar hasta 8 valores.

Octal

Cada dígito en esta representación puede representar hasta 8 valores.



Octal

Cada dígito en esta representación puede representar hasta 8 valores.



0
1
2
3
4
5
6
7
10
11
12
13
14
15
16
17
20
21
...
77
100

0

1

2

3

4

5

6

7

10

11

12

13

14

15

16

17

20

21

...

77

100

Comenzamos el conteo

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 20
- 21
- ...
- 77
- 100

Sumamos 1

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 20
- 21
- ...
- 77
- 100

Sumamos 1

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 20
- 21
- ...
- 77
- 100

Sumamos 1

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 20
- 21
- ...
- 77
- 100

Sumamos 1

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 20
- 21
- ...
- 77
- 100

Sumamos 1

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 20
- 21
- ...
- 77
- 100

Sumamos 1

0
1
2
3
4
5
6
7
10
11
12
13
14
15
16
17
20
21
...
77
100

Alcanzamos el máximo valor por dígito

0
1
2
3
4
5
6
7
10
11
12
13
14
15
16
17
20
21
...
77
100

Comenzamos de nuevo desde 10

0
1
2
3
4
5
6
7
10
11
12
13
14
15
16
17
20
21
...
77
100

Sumamos 1 hasta llegar a 17

0
1
2
3
4
5
6
7
10
11
12
13
14
15
16
17
20
21
...
77
100

Comenzamos desde 20

0
1
2
3
4
5
6
7
10
11
12
13
14
15
16
17
20
21
...
77
100

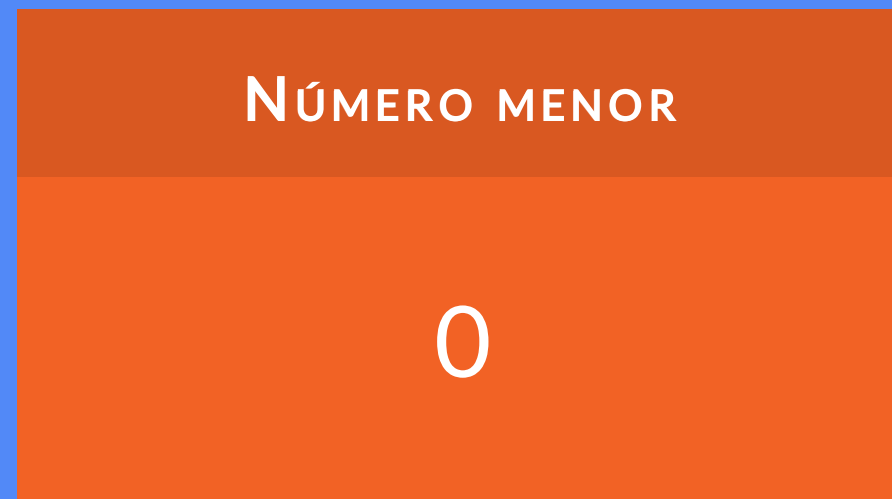
Y así sucesivamente hasta que llegamos al 77, y de ahí pasamos al 100

Hexadecimal

Cada dígito en esta representación puede representar hasta 15 distintos valores. Para esto, se utilizan los números del 0-1, y las letras de la A-F.

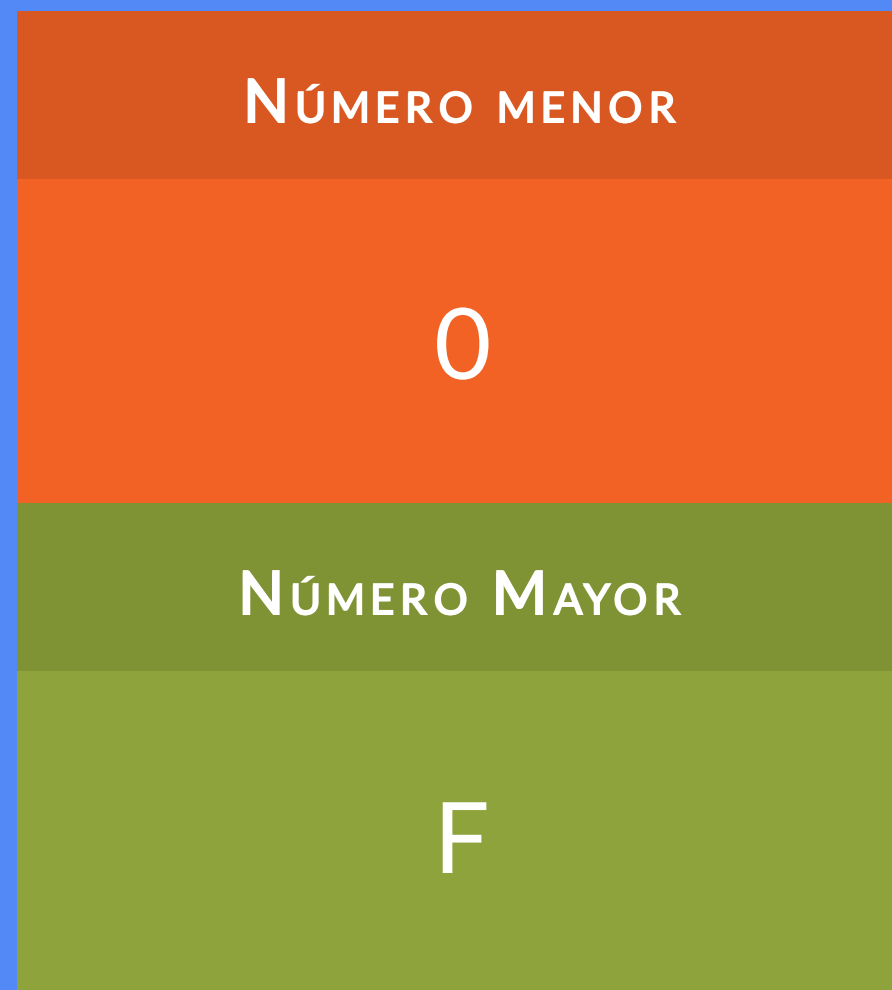
Hexadecimal

Cada dígito en esta representación puede representar hasta 15 distintos valores. Para esto, se utilizan los números del 0-1, y las letras de la A-F.



Hexadecimal

Cada dígito en esta representación puede representar hasta 15 distintos valores. Para esto, se utilizan los números del 0-1, y las letras de la A-F.



Hexadecimal

En ciencias computacionales, la base decimal la vemos representada con un prefijo "0x":

0xF3
0x6A23

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- A
- B
- C
- D
- E
- F
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18

- 8
- 9
- A
- B
- C
- D
- E
- F
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 1A
- 1B
- 1C
- 1D
- 1E
- 1F
- 20

Binario a Octal

Dado el siguiente número binario:

$$100111_2 \rightarrow x_8$$

Binario a Octal

Dado el siguiente número binario:

$$100111_2 \rightarrow x_8$$

1. De derecha a izquierda, agrupamos grupos de 3 bits.

$$\boxed{100} \boxed{111}_2 \rightarrow x_8$$

Binario a Octal

Dado el siguiente número binario:

$$100111_2 \rightarrow x_8$$

1. De derecha a izquierda, agrupamos grupos de 3 bits.

$$\boxed{100} \boxed{111}_2 \rightarrow x_8$$

2. Convertimos cada sección a octal. Recuerda que cada posición puede tener valores del 0 al 7.

$$\boxed{100}_2 = 4$$

$$\boxed{111}_2 = 7$$

Binario a Octal

Dado el siguiente número binario:

$$100111_2 \rightarrow x_8$$

1. De derecha a izquierda, agrupamos grupos de 3 bits.

$$\boxed{100} \boxed{111}_2 \rightarrow x_8$$

2. Convertimos cada sección a octal. Recuerda que cada posición puede tener valores del 0 al 7.

$$\boxed{100}_2 = 4$$

$$\boxed{111}_2 = 7$$

3. Juntamos los números.

$$100111_2 = 47_8$$

Binario a Hexadecimal

Dado el siguiente número binario:

$$110000101101_2 \rightarrow x_{16}$$

Binario a Hexadecimal

Dado el siguiente número binario:

$$110000101101_2 \rightarrow x_{16}$$

1. De derecha a izquierda, agrupamos grupos de 4 bits.

$$\boxed{1100} \boxed{0010} \boxed{1101}_2 \rightarrow x_{16}$$

Binario a Hexadecimal

Dado el siguiente número binario:

$$110000101101_2 \rightarrow x_{16}$$

1. De derecha a izquierda, agrupamos grupos de 4 bits.

$$\boxed{1100} \boxed{0010} \boxed{1101}_2 \rightarrow x_{16}$$

2. Convertimos cada sección a hexadecimal. Recuerda que cada posición puede tener valores del 0 al F.

$$\boxed{1100}_2 = 12_{10} = C_{16}$$

$$\boxed{0010}_2 = 2_{10} = 2_{16}$$

$$\boxed{1101}_2 = 13_{10} = D_{16}$$

Binario a Hexadecimal

Dado el siguiente número binario:

$$110000101101_2 \rightarrow x_{16}$$

1. De derecha a izquierda, agrupamos grupos de 4 bits.

$$\boxed{1100} \boxed{0010} \boxed{1101}_2 \rightarrow x_{16}$$

2. Convertimos cada sección a hexadecimal. Recuerda que cada posición puede tener valores del 0 al F.

$$\boxed{1100}_2 = 12_{10} = C_{16}$$

$$\boxed{0010}_2 = 2_{10} = 2_{16}$$

$$\boxed{1101}_2 = 13_{10} = D_{16}$$

3. Juntamos los números.

$$110000101101_2 = C2D_{16}$$

Conversiones de Hexadecimal a Decimal

Dado un número en hexadecimal:

$$23E_{16}$$

Conversiones de Hexadecimal a Decimal

Dado un número en hexadecimal:

$$23E_{16}$$

1. Extendemos el polinomio a través de multiplicar cada dígito *convertido a decimal* por la base origen elevado a su posición.

Recordemos que: $E_{16} = 14_{10}$

$$23E_{16} = 2 \cdot 16^2 + 3 \cdot 16^1 + 14 \cdot 16^0$$

$$23E_{16} = 512_{10} + 48_{10} + 14_{10}$$

Conversiones de Hexadecimal a Decimal

Dado un número en hexadecimal:

$$23E_{16}$$

1. Extendemos el polinomio a través de multiplicar cada dígito *convertido a decimal* por la base origen elevado a su posición.

Recordemos que: $E_{16} = 14_{10}$

$$23E_{16} = 2 \cdot 16^2 + 3 \cdot 16^1 + 14 \cdot 16^0$$

$$23E_{16} = 512_{10} + 48_{10} + 14_{10}$$

2. Completando la suma

$$23E_{16} = 574_{10}$$

**Este mismo
procedimiento puede
ser usado para
convertir cualquier
base a decimal!**

Codificación de Información

Codificación

La codificación de caracteres es el método que permite convertir un carácter de un alfabeto en un símbolo de otro sistema de representación, aplicando normas o reglas de codificación. Para representar información en las computadoras, utilizamos las siguientes codificaciones:



Codificación

La codificación de caracteres es el método que permite convertir un carácter de un alfabeto en un símbolo de otro sistema de representación, aplicando normas o reglas de codificación. Para representar información en las computadoras, utilizamos las siguientes codificaciones:



Codificación

La codificación de caracteres es el método que permite convertir un carácter de un alfabeto en un símbolo de otro sistema de representación, aplicando normas o reglas de codificación. Para representar información en las computadoras, utilizamos las siguientes codificaciones:

- ASCII



Codificación

La codificación de caracteres es el método que permite convertir un carácter de un alfabeto en un símbolo de otro sistema de representación, aplicando normas o reglas de codificación. Para representar información en las computadoras, utilizamos las siguientes codificaciones:

- ASCII
- UTF-8



Codificación

La codificación de caracteres es el método que permite convertir un carácter de un alfabeto en un símbolo de otro sistema de representación, aplicando normas o reglas de codificación. **Para representar información en las computadoras, utilizamos las siguientes codificaciones:**

- ASCII
- UTF-8
- Unicode



Codificación

Podemos expresar la cantidad de caracteres representables "M" en función de la cantidad de bits "n" con la siguiente relación.

$$M(n) = 2^n$$

Por lo que con un byte, se puede representar:

$$M(8) = 2^8 = 256$$

Codificación

Codificación

Podemos despejar la ecuación para obtener la cantidad de bits mínimos para representar un alfabeto, por ejemplo:

$$n \geq \log_2(M)$$
$$n \geq 3.32 \log(M)$$

Codificación

Podemos despejar la ecuación para obtener la cantidad de bits mínimos para representar un alfabeto, por ejemplo:

$$n \geq \log_2(M)$$
$$n \geq 3.32 \log(M)$$

Por ejemplo, para codificar el siguiente alfabeto:

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Codificación

Podemos despejar la ecuación para obtener la cantidad de bits mínimos para representar un alfabeto, por ejemplo:

$$\begin{aligned}n &\geq \log_2(M) \\n &\geq 3.32\log(M)\end{aligned}$$

Por ejemplo, para codificar el siguiente alfabeto:

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Necesaríamos:

$$\begin{aligned}n &\geq 3.32\log(10) \\n &\geq 3.32\end{aligned}$$

Codificación

Podemos despejar la ecuación para obtener la cantidad de bits mínimos para representar un alfabeto, por ejemplo:

$$\begin{aligned}n &\geq \log_2(M) \\n &\geq 3.32\log(M)\end{aligned}$$

Por ejemplo, para codificar el siguiente alfabeto:

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Necesaríamos:

$$\begin{aligned}n &\geq 3.32\log(10) \\n &\geq 3.32\end{aligned}$$

Redondeado sería

$$n \geq 4$$

ASCII enco

ASCII es una codificación que requiere **1 byte por caracter**. La codificación ASCII incluye:

5	5	005	ENQ	(enquiry)	37	25	045	##37;	%	69	45	105	##69;	E	101	65	145	##101;	e
6	6	006	ACK	(acknowledge)	38	26	046	##38;	&	70	46	106	##70;	F	102	66	146	##102;	f
7	7	007	BEL	(bell)	39	27	047	##39;	'	71	47	107	##71;	G	103	67	147	##103;	g
8	8	010	BS	(backspace)	40	28	050	##40;	(72	48	110	##72;	H	104	68	150	##104;	h
9	9	011	TAB	(horizontal tab)	41	29	051	##41;)	73	49	111	##73;	I	105	69	151	##105;	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	##42;	*	74	4A	112	##74;	J	106	6A	152	##106;	j
11	B	013	VT	(vertical tab)	43	2B	053	##43;	+	75	4B	113	##75;	K	107	6B	153	##107;	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	##44;	,	76	4C	114	##76;	L	108	6C	154	##108;	l
13	D	015	CR	(carriage return)	45	2D	055	##45;	-	77	4D	115	##77;	M	109	6D	155	##109;	m
14	E	016	SO	(shift out)	46	2E	056	##46;	.	78	4E	116	##78;	N	110	6E	156	##110;	n
15	F	017	SI	(shift in)	47	2F	057	##47;	/	79	4F	117	##79;	O	111	6F	157	##111;	o
16	10	020	DLE	(data link escape)	48	30	060	##48;	0	80	50	120	##80;	P	112	70	160	##112;	p
17	11	021	DC1	(device control 1)	49	31	061	##49;	1	81	51	121	##81;	Q	113	71	161	##113;	q
18	12	022	DC2	(device control 2)	50	32	062	##50;	2	82	52	122	##82;	R	114	72	162	##114;	r
19	13	023	DC3	(device control 3)	51	33	063	##51;	3	83	53	123	##83;	S	115	73	163	##115;	s
20	14	024	DC4	(device control 4)	52	34	064	##52;	4	84	54	124	##84;	T	116	74	164	##116;	t
21	15	025	NAK	(negative acknowledge)	53	35	065	##53;	5	85	55	125	##85;	U	117	75	165	##117;	u
22	16	026	SYN	(synchronous idle)	54	36	066	##54;	6	86	56	126	##86;	V	118	76	166	##118;	v
23	17	027	ETB	(end of trans. block)	55	37	067	##55;	7	87	57	127	##87;	W	119	77	167	##119;	w
24	18	030	CAN	(cancel)	56	38	070	##56;	8	88	58	130	##88;	X	120	78	170	##120;	x
25	19	031	EM	(end of medium)	57	39	071	##57;	9	89	59	131	##89;	Y	121	79	171	##121;	y
26	1A	032	SUB	(substitute)	58	3A	072	##58;	:	90	5A	132	##90;	Z	122	7A	172	##122;	z
27	1B	033	ESC	(escape)	59	3B	073	##59;	;	91	5B	133	##91;	[123	7B	173	##123;	{
28	1C	034	FS	(file separator)	60	3C	074	##60;	<	92	5C	134	##92;	\	124	7C	174	##124;	
29	1D	035	GS	(group separator)	61	3D	075	##61;	=	93	5D	135	##93;]	125	7D	175	##125;	}
30	1E	036	RS	(record separator)	62	3E	076	##62;	>	94	5E	136	##94;	^	126	7E	176	##126;	~
31	1F	037	US	(unit separator)	63	3F	077	##63;	?	95	5F	137	##95;	_	127	7F	177	##127;	DEL

Source: www.LookupTables.com

ASCII encoding

ASCII es una codificación que requiere **1 byte por caracter**. La codificación ASCII incluye:

- 128 caracteres de ASCII estándar

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

ASCII encoding

ASCII es una codificación que requiere **1 byte por caracter**. La codificación ASCII incluye:

- 128 caracteres de ASCII estándar
- 128 caracteres de ASCII extendido

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Unicode

Es mantenido por **The Unicode Consortium**. Hasta su versión 12.1, contiene un total de **137,929** caracteres.

Incluye:

- Caracteres A U+0041
- Íconos 🤖 U+2602
- Modificadores ☒☒☒☒☒☒☒ U+1F476 + U+1F3FB
- Kanji あ U + 3041
- Emojis 😊 U + 1F60A

Unicode

C0 Controls and Basic Latin																
U+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000x	[NUL]	[SOH]	[STX]	[ETX]	[EOT]	[ENQ]	[ACK]	[BEL]	[BS]	[HT]	[LF]	[VT]	[FF]	[CR]	[SO]	[SI]
001x	[DLE]	[DC1]	[DC2]	[DC3]	[DC4]	[NAK]	[SYN]	[ETB]	[CAN]	[EM]	[SUB]	[ESC]	[FS]	[GS]	[RS]	[US]
002x	[SP]	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
003x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
006x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	[DEL]
C1 Controls and Latin-1 Supplement																
U+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
008x	[XXX]	[XXX]	[BPH]	[NBH]	[IND]	[NEL]	[SSA]	[ESA]	[HTS]	[HTJ]	[VTS]	[PLD]	[PLU]	[RI]	[SS2]	[SS3]
009x	[DCS]	[PU1]	[PU2]	[STS]	[CCH]	[MW]	[SPA]	[EPA]	[SOS]	[XXX]	[SCI]	[CSI]	[ST]	[OSC]	[PM]	[APC]
00Ax	[NB SP]	ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	[SHY -]	®	¯
00Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ