# SISTEMAS DE NUMERACIÓN

[SISTEMAS DE NUMERACIÓN](#_Toc152105832)

[1.- Empleo de Códigos a través de la historia](#_Toc152105833)

[2.- Los Bits y los Bytes](#_Toc152105834)

[2.1.- Así funcionan los bits y los bytes en el ordenador.](#_Toc152105835)

[2.2.- Múltiplos de byte](#_Toc152105836)

[2.3.- Base de un sistema numérico](#_Toc152105837)

[3.- Sistemas de numeración](#_Toc152105838)

[3.1.- El sistema decimal](#_Toc152105839)

[3.2.- El sistema binario](#_Toc152105840)

[3.3.- El sistema hexadecimal](#_Toc152105841)

[3.4.- El Sistema Octal](#_Toc152105842)

[3.5.- Operaciones con binarios](#_Toc152105843)

[3.5.1.- Suma de números binarios](#_Toc152105844)

[3.5.2.- Resta de números binarios](#_Toc152105845)

[3.5.3.- Producto de números binarios](#_Toc152105846)

[3.6.- Código binario, decimal ,octal y hexadecimal](#_Toc152105847)

[3.6.1.- De binario a decimal](#_Toc152105848)

[3.6.2.- De decimal a binario](#_Toc152105849)

[3.6.3.- De octal a decimal](#_Toc152105850)

[3.6.4.- De decimal a octal](#_Toc152105851)

[3.6.5.- De octal a binario](#_Toc152105852)

[3.6.6.- De binario a octal](#_Toc152105853)

[3.6.7.- De hexadecimal a decimal](#_Toc152105854)

[3.6.8.- De hexadecimal a binario](#_Toc152105855)

[3.6.8.- De binario a hexadecimal](#_Toc152105856)

# 1.- Empleo de Códigos a través de la historia

Para establecer una analogía entre el ordenador, computadora personal o PC  y otros sistemas de comunicación por código, se puede decir que éste no fue el primer dispositivo en utilizar ese recurso.

De hecho, mucho antes de que existiera algo parecido a los ordenadores, las comunidades primitivas transmitían mensajes a largas distancias utilizando **códigos de sonidos** (por medio de tambores, cuernos) o **visuales** (produciendo señales de humo). Incluso las marinas de guerra de algunos países todavía utilizan un antiguo **código de banderas** llamado semáforo para transmitir mensajes entre buques que se encuentran a la vista en alta mar, o entre buques y tierra

Otro ejemplo lo tenemos en el código o alfabeto **Morse** de **telegrafía**. Lo creó en 1838 [**Samuel Morse**](http://www.asifunciona.com/biografias/morse/morse.htm) (inventor a su vez del telégrafo) y se utilizó masivamente hasta hace pocos años para el envío de mensajes por cables o por vía inalámbrica a todo el mundo, utilizando únicamente **puntos y rayas, a modo de código binario analógico**. Por medio de una llave telegráfica Morse, se transmitían y enviaban los mensajes a través de un tendido de alambres de cobre. En otro punto distante se recibían dichos mensajes en un dispositivo receptor electromagnético en forma de sonidos monorrítmicos, cortos y largos, que representaban las letras, números y signos.

# 2.- Los Bits y los Bytes

Seguramente conoces que un ordenador constituye un dispositivo electrónico digital. La palabra “**digital**” está relacionada con el término “**dígito**”, que a su vez significa “**dedo**”.

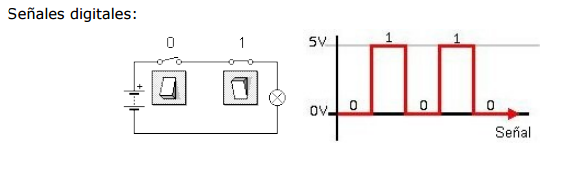
La etimología de esta palabra proviene de la época en que nuestros **antepasados** tenían que **contar con los dígitos o dedos** las piezas que cazaban. De ahí también que las impresiones que dejan nuestros dedos cuando tocamos un objeto se denominen “huellas digitales”.

## 2.1.- Así funcionan los bits y los bytes en el ordenador.

Por un acuerdo conjunto entre los ingenieros y científicos, se le asignó al **dígito “1” la existencia de un pulso** eléctrico y **al dígito “0” la no existencia de pulso eléctrico**. Por tanto, para el ordenador sólo existen **dos estados físicos** que le permiten comprender las órdenes o instrucciones antes de ejecutarlas: “la existencia de pulsos eléctricos o la no existencia de ellos”.

Por ejemplo, cuando se escribe en el teclado la letra **“A” mayúscula,** se generan **automáticamente 8 bits u octeto**, equivalentes a un byte, que representan esa letra. El código numérico que se genera, para que el ordenador reconozca que se ha escrito la letra “A” , es: 0100 0001. Cada uno de los bits correspondientes a los dígitos “1” contenidos en ese byte de información generan pulsos eléctricos, mientras que los representados por el dígito “0” no generan prácticamente ningún pulso eléctrico.

En cualquier circuito electrónico digital, como el que posee el ordenador, el bit “0” puede estar en ocasiones cercano a “0” volt y el **bit “1” cercano a 3 ó 5 volt,** de forma tal que la tensión o voltaje que pueda llegar a tener el dígito “0” nunca llegará a alcanzar un valor alto, ni el dígito “1” un valor muy bajo.

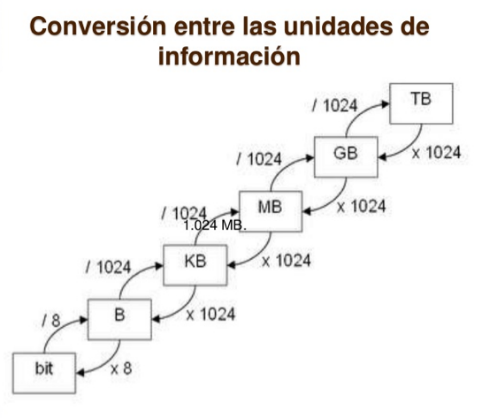


Gracias a ese mecanismo el circuito digital puede diferenciar perfectamente el valor correspondiente a estos dos dígitos sin equivocarse, por lo que el riesgo de que se produzcan confusiones o errores a la hora de reconocer el valor de ambos es prácticamente nula.

## 2.2.- Múltiplos de byte

La capacidad de almacenamiento de la memoria RAM y de los dispositivos empleados para almacenar programas, documentos de texto, datos, música, fotos e imágenes en movimiento se mide también en “bytes”. Pero cuando se trata de grandes cantidades de bytes contenidas en un archivo o en una carpeta incluida dentro de un dispositivo de almacenamiento masivo de información, como puede ser un disquete, disco duro, CD, DVD, etc., se utilizan los siguientes múltiplos del byte:

* 1 bit
* 1 nibble = 4 bits
* **1 byte = 1 octeto = 8 bits**
* kilobyte (kB) = 1.024 bytes = 210
* megabyte (MB)=1.024 kB= 1024\*1024= 1 048 576 bytes = 210. 210 = 220
* gigabyte (GB) = 1.024MB = 1024\*1024\*1024= 1 073 741 824 bytes
* terabyte (TB) = 1.024GB= 1024\*1024\*1024\*1024=1 099 511 627 776 bytes



Por eso, cuando queremos **adquirir** un **ordenador** o computadora personal, además de interesarnos por el tipo de **microprocesador** que utiliza, debemos preocuparnos también por la **capacidad de almacenamiento** de datos en gibabytes (GB) o terabytes que admiten tanto la memoria principal de trabajo (**RAM**), como el disco duro.

## 2.3.- Base de un sistema numérico

La **base** de un sistema numérico radica en la cantidad de **dígitos** diferentes que son **necesarios** para **representar** las cifras.

Por ejemplo, a continuación, se puede apreciar la cantidad de dígitos diferentes que emplea un sistema numérico en particular, de acuerdo con su correspondiente base numérica

|  |  |  |
| --- | --- | --- |
| **BASE NUMÉRICA** | **DÍGITOS EMPLEADOS** | **CANTIDAD TOTAL DE DÍGITOS** |
| Binaria(2) | 0 y 1 | 2 |
| Octal(8) | 0, 1, 2, 3, 4, 5, 6 y 7 | 8 |
| Decimal(10) | 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9 | 10 |
| Hexadecimal(16) | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F | 16 |

# 3.- Sistemas de numeración

|  |  |  |
| --- | --- | --- |
| Se supone que por la necesidad que tenían los primeros “homo sapiens” de utilizar los diez dedos de las **manos** para **contar** (recurso que aún utilizan muchos niños e incluso no muy niños… :P), surgió el sistema numérico que aprendemos desde muy temprano en la escuela, compuesto por diez dígitos o números que van del “0” al “9”. Ese sistema que todos conocemos, se denomina "**sistema numérico decimal**", o "de base 10". |  |  |

Todo número se expresa con un conjunto de cifras, contribuyendo cada una de ellas con un valor que depende:

1. de la **cifra** en sí,
2. de la **posición** que ocupa dentro del número

## 3.1.- El sistema decimal

El sistema decimal, el más utilizado en todos los ámbitos de la actividad humana, se distingue por las siguientes características:

* Utiliza una base 10.
* Sus numerales son las cifras del 0 al 9, ambas incluidas.
* Las **posiciones** relativas de los números se denominan unidades, decenas, centenas, unidades de millar, decenas de millar, centenas de millar, unidades de millón, etc.

Por ejemplo, el número decimal 278,5 puede obtenerse como suma de :

200

70

8 10\*10 10 1

0,5

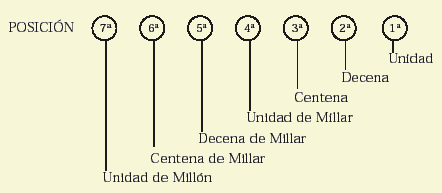
------------

275,5

Es decir, se verifica que :

278,5 = 2 x 10**2** + 7 x 10**1** + 8 x 10**0** + 5 x 10**-1**

Cada **posición**, por tanto, tiene un **peso** específico. En el sistema decimal, cada posición tiene un nombre. 🡪 **Sistema posicional**



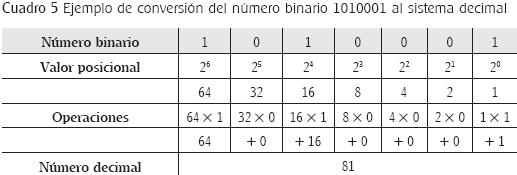
Pero en el mundo de las matemáticas el sistema decimal no es único que existe para realizar cálculos simples o complejos. Coexisten, además, otros sistemas numéricos, prácticamente desconocidos para la mayoría de la gente, entre los que se encuentran el **"sistema numérico hexadecimal",** de "**base 16",** y el "**sistema numérico binario**", de "**base 2".**

## 3.2.- El sistema binario

Este último es el más utilizado en informática y emplea para efectuar todas las operaciones matemáticas solamente el “0” y “1”, dígitos con los cuales los **ordenadores** realizan **todas las operaciones** para las que fueron concebidos. De ahí su denominación de "dispositivos **digitales**".

El sistema numérico binario fue el escogido por los ingenieros informáticos para el funcionamiento de los ordenadores, porque era **más fácil para el sistema electrónico** de la máquina distinguir y manejar solamente dos dígitos, o sea, el "0" y el "1" que componen el sistema numérico binario, en lugar de los diez dígitos (del 0 al 9), que constituyen el sistema numérico decimal.

Todos los **programas, instrucciones, textos y órdenes** que introducimos en el ordenador éste las recibe en **código binario** como una cadena de ceros y unos. Cada cero (“0”) y cada uno (“1”), representa un “**bit**” de información. La palabra “bit” constituye el acrónimo de Binary DigIT, que significa “dígito binario”.



**Números binarios del 0 al 7**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| binario | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| decimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

1x 2**2** + 1x 2**1** + 1x 2**0**

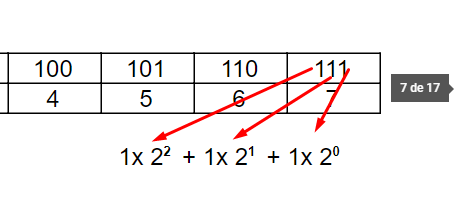
****

Imagen que contiene texto

Descripción generada automáticamente

Para formar cada carácter alfanumérico, es decir una letra, número o signo, los ingenieros informáticos, después de realizar muchas pruebas, optaron por combinar **ocho bits** o cadena de ceros y unos para formar un “**octeto**” al que denominaron “**byte”**.

## 3.3.- El sistema hexadecimal

Nuestro sistema decimal regular tiene como base a las decenas, utilizando diez símbolos diferentes para mostrar los números. El **hexadecimal** es un sistema que tiene como **base** el **dieciséis**, lo que significa que emplea dieciséis caracteres para mostrar los números.

Con el auge de la informática y de la lógica binaria, el sistema de numeración hexadecimal experimentó un gran auge al ser utilizado en los programas y los ordenadores. Este sistema tiene base 16. Los dígitos que utiliza normalmente son las cifras del 0 al 9 y, después, las letras A, B, C, D, E y F para denotar los guarismos(Signo gráfico que expresa un número en un sistema de numeración) décimo a decimoquinto.

Para representar un número en base hexadecimal (esto es, b=16) es necesario disponer de un alfabeto de 16 símbolos: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, **A, B, C, D, E, F**}

Tabla 4.2.-Números binarios del 0 al 7

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

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

La conversión entre números hexadecimales y binarios se realiza "expandiendo" o "con­trayendo" **cada dígito** hexadecimal a **cuatro dígitos binarios**. Por ejemplo, para expresar en hexadecimal el número binario 1010011100112 bastará con tomar grupos de cuatro bits, **empezando por la derecha**, y reemplazarlos por su equivalente hexadecimal:

**10102 = A16**

0111**2 = 716**

**00112 = 316**

y, por tanto: **1010011100112 = A7316**

En caso de que los dígitos binarios no formen grupos completos de cuatro dígitos, **se deben añadir ceros a la izquierda hasta completar** el último grupo. Por ejemplo:

**1011102 = 001011102 = 2E16**

Pasar a decimal: 10011001 -------------- 11110010 ------- 01101101 ------ 11101

## 3.4.- El Sistema Octal

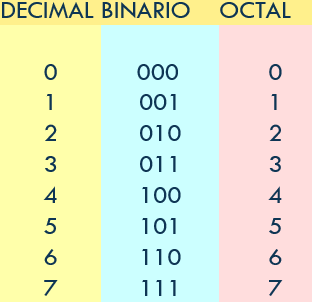
El [sistema de numeración](https://es.wikipedia.org/wiki/Sistema_de_numeraci%C3%B3n) posicional cuya base es [8](https://es.wikipedia.org/wiki/Ocho), se llama **octal** y utiliza los dígitos indio arábigos: **0,1,2,3,4,5,6,7.**

Los números octales son formados a partir de los números binarios.

Esto es así porque su **base** es una [potencia](https://www.lifeder.com/potencia-fisica/) exacta de dos (8=23). Es decir, los números que pertenecen al sistema octal se forman cuando estos son agrupados en **tres dígitos** consecutivos, ordenados de derecha a izquierda, obteniendo de esa forma su valor decimal.

Las posiciones de los dígitos en un número octal tienen los siguientes pesos:

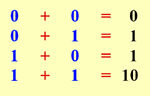
84, 83, 82, 81, 80, punto octal, 8-1, 8-2, 8-3, 8-4, 8-5.



## 3.5.- Operaciones con binarios

### 3.5.1.- Suma de números binarios

**Tabla para sumar binarios**



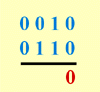
De la misma forma que hacemos cuando sumamos números del sistema decimal, esta operación matemática la comenzamos a realizar de derecha a izquierda, comenzando por los últimos dígitos de ambos sumandos, como en el siguiente ejemplo en el que sumaremos:

**0010**

**+ 0110**

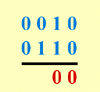
**Primer paso**

De la misma forma que hacemos cuando sumamos números del sistema decimal, esta operación matemática la comenzamos a realizar de derecha a izquierda, comenzando por los últimos dígitos de ambos sumandos, como en el siguiente ejemplo:



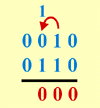
**Segundo paso**

Se suman los siguientes dígitos 1 + 1 = 10 (según la tabla), se escribe el “0” y se acarrea o lleva un “1”. Por tanto, el “0” correspondiente a tercera posición de izquierda a derecha del primer sumando, adquiere ahora el valor “1



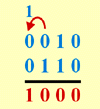
**Tercer paso**

Al haber tomado el “0” de la tercera posición el valor “1”, tendremos que sumar 1 + 1 = 10. De nuevo acarreamos o llevamos un “1”, que tendremos que pasar a la cuarta posición del sumando.



**Cuarto paso**

El valor “1” que toma el dígito “0” de la cuarta posición lo sumamos al dígito “0” del sumando de abajo. De acuerdo con la tabla tenemos que 1+ 0 = 1.



11100110 + 00111101

El resultado final de la suma de los dos números binarios será: **1 0 0** **0**.

### 3.5.2.- Resta de números binarios

Cuando se quiere restar entre números binarios, se usa el mismo método que en el sistema decimal, con la misma idea de “llevar uno” que en la suma, pero la diferencia es que al llevar uno, **el número que sobra se debe restar en la siguiente columna** o posición de la cifra binaria, y la tabla de restas viene a ser parecida pero con una diferencia.

**Tabla para restar binarios**

**0** **–**  **0 =** **0**  
**1**  **–**  **1** **=** **0**  
**1 –**  **0** **=** **1**

**0**  **– 1 = 1** (con acarreo negativo de 1 - se transforma en 10 - 1 = 1)

No se puede restar **0**  **– 1** de la forma tradicional, pues en el sistema binario, los números negativos tienen un método distinto para ser representados y en una sola cifra no puede existir un número negativo de forma unitaria, sino que toda la cifra deberá ser convertida una vez que se la obtenga.

Restar **011 de 101.**

Solución.  **1 0 1 5**

**1**

**- 0 1 1 - 3**

**---------- -----**

**0 1 0 2**

En este ejemplo es necesario un acarreo negativo.

Comenzando por la columna de la derecha, se tiene:

**110 1**

**- 0 1 1**

**-----------**

**0 1 0**

### 3.5.3.- Producto de números binarios

**Tabla para multiplicar binarios**

**0** **x**  **0 =** **0**  
**1**  **x** **1** **=** **1**  
**1 x**  **0** **=** **0**

**0**  **x 1 = 0**

El algoritmo del [producto](http://es.wikipedia.org/wiki/Multiplicaci%C3%83%C2%B3n) en binario es igual que en números decimales; aunque se lleva cabo con más sencillez, ya que el **0 multiplicado por cualquier número da 0**, y el **1** es el [**elemento neutro**](http://es.wikipedia.org/wiki/Elemento_neutro) del producto.

Por ejemplo, multipliquemos 10110 por 1001:

**10110         
     x  1001                      
   —————————            
        10110                 
       00000                  
     00000**

**10110                  
   —————————             
  11000110**

**1111 \* 111**

## 3.6.- Código binario, decimal ,octal y hexadecimal

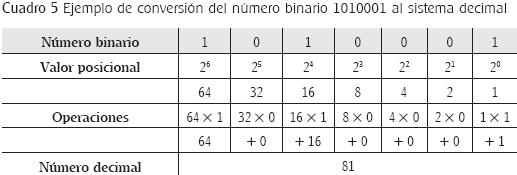
### 3.6.1.- De binario a decimal

En sistema decimal, las cifras que componen un número son las cantidades que están multiplicando a las distintas potencias de diez (10, 100, 1000, 10000, etc.)

Por ejemplo, **745** = **7** · 100 + **4** · 10 + **5** · 1

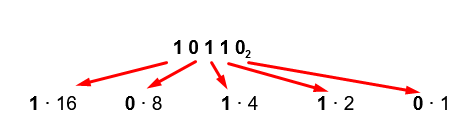
O lo que es lo mismo: **745** = **7** · 102 + **4** · 101 + **5** · 100

En el **sistema binario**, las cifras que componen el número multiplican a las potencias de dos (1, 2, 4, 8, 16, ….)



**26=64, 25=32, 24=16, 23=8, 22=4, 21=2, 20=1**

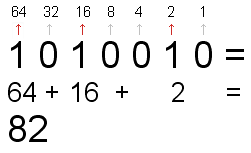
Por ejemplo, para pasar un número binario a decimal, empezamos por la derecha y vamos multiplicando cada cifra por las sucesivas potencias de 2, avanzando hacia la izquierda:



16  + 4 + 2 = **2210**

**1102** = **0** · 1 + **1** · 2 + **1** · 4 = 2 + 4 = **610**

**Procedimiento simplificado**:-Asignamos a cada dígito su valor-Seleccionamos los que valgan 1-Sumamos



### 3.6.2.- De decimal a binario

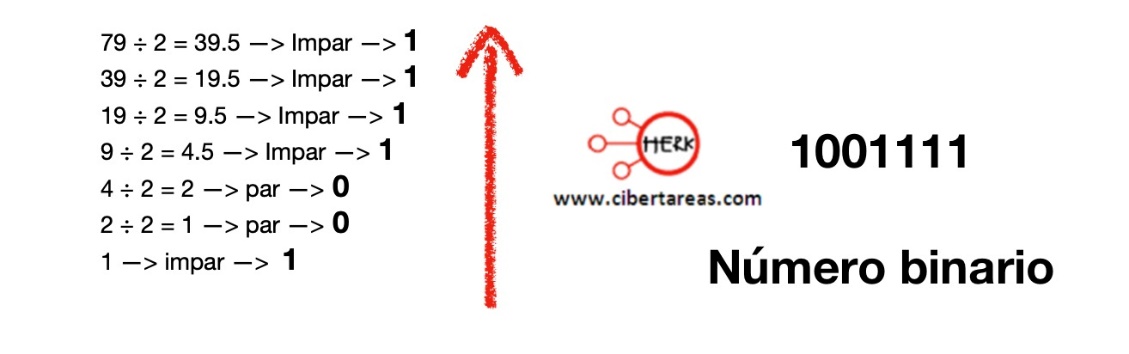
Para hacer la conversión de decimal a binario, hay que **ir dividiendo el número decimal entre dos** y anotar en una columna a la derecha el resto (un 0 si el resultado de la división es par y un 1 si es impar).

La lista de ceros y unos leídos de abajo a arriba es el resultado.

Ejemplo: vamos a pasar a binario 7910

79**1**(impar). Dividimos entre dos:  
39**1** (impar). Dividimos entre dos:  
19**1**(impar). Dividimos entre dos:  
9**1**(impar). Dividimos entre dos:  
4**0** (par). Dividimos entre dos:  
2**0** (par). Dividimos entre dos:  
1     **1** (impar).

Por tanto**, 7910 = 1001111**



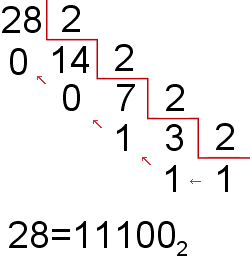
**Procedimiento**:

Se aplica el MDS (**método** de las **divisiones sucesivas**)

- **Dividir entre 2 sucesivamente**

- Apuntar el resultado y el resto de cada operación

- Apuntar a lista de ceros y unos **de abajo a arriba**

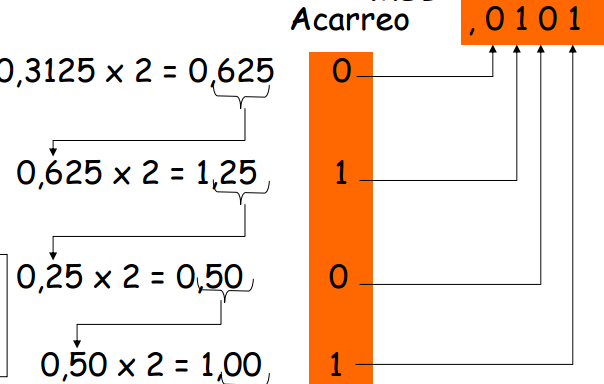


Cuando el número contiene decimales estos se multiplican sucesivamente por 2.

Por ejemplo, para convertir en binario el número fraccionado **0.3125** empezamos multiplicando por 2, y después se **multiplica** cada **parte fraccional** resultante del producto por 2, **hasta** que el **producto** fraccionario sea **cero** o hasta que se alcance el número deseado de posiciones decimales.

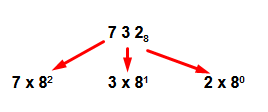
Los dígitos acarreados o acarreos generados por las multiplicaciones dan lugar al nº binario.

Ejemplo:

****

### 3.6.3.- De octal a decimal

Para convertir un número del sistema octal a su equivalente en el sistema decimal solo se tiene que **multiplicar** cada **dígito octal** por su valor **posicional**, comenzando desde la **derecha**.



|  |  |  |
| --- | --- | --- |
| **82** | **81** | **80** |
| **64** | **8** | **1** |
| 7 | 3 | 2 |

**7328 = (7 \* 64) + (3 \* 8) + (2 \* 1)**

**7328= 448 +24 +2**

**7328= 47410**

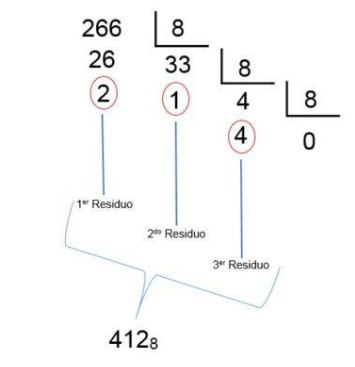
### 3.6.4.- De decimal a octal

Un número entero decimal puede ser convertido en un número octal utilizando el método de la **división repetida**, donde el entero **decimal se divide entre 8** **hasta** que el **cociente** sea **igual a 0**, y los residuos de cada división van a representar al número octal.

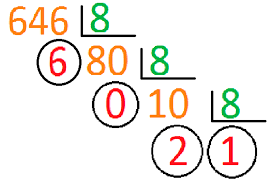
Los **residuos** son ordenados **del último al primero**; es decir, que el primer residuo será el dígito menos significativo del número octal. De esa forma, el dígito más significativo será el último resto.

**Ejemplo**

Octal del número decimal: **26610**

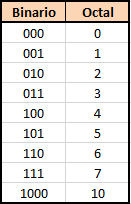


Decimal 646🡪 a OCTAL 1206



### 3.6.5.- De octal a binario

La conversión del sistema octal al binario se lleva a cabo al convertir el dígito octal a su dígito binario equivalente, formado por **tres dígitos(1 digito octal equivale a 3 en binario)**. Existe una tabla que muestra cómo se convierten los ocho posibles dígitos:



A partir de esas conversiones se puede cambiar cualquier número del sistema octal al binario, como por ejemplo, para convertir el número 5728 se buscan sus equivalentes en la tabla. Así, se tiene que:

58 = **101**

78=111

28 = **010**

Por lo tanto, 5728 equivale en el sistema binario a **101** 111 **010**.

### 3.6.6.- De binario a octal

El proceso de conversión de números enteros binarios a números enteros octales es la operación inversa al proceso anterior.

Es decir, **se agrupan los bits** del número binario en dos **grupos de tres bits**, comenzando de derecha a izquierda. Luego, se hace la conversión de binario a octal con la tabla anterior.

En algunos casos el número binario no tendrá grupos de **3 bits**; para completarlo, se **agregan** uno o dos **ceros** **a la izquierda del primer** grupo.

Por ejemplo, para cambiar el número binario **11010110** a **octal** se realiza lo siguiente:

– Se forman grupos de 3 bits comenzando por la derecha (último bit):

11 010 110

– Como el primer grupo está incompleto, se agrega un cero a la izquierda:

**0**11 010 110

– Se hace la conversión a partir de la tabla:

**011** = **3**

010 = 2

**110 = 6**

De esa forma, el número binario **011**010**110** equivale a **3**2**6**8.

### 3.6.7.- De hexadecimal a decimal

Consiste en utilizar las letras A, B, C, D, E y F para representar los números del diez al quince, 10,11,12,13,14,15

mientras que para el dieciséis emplearemos el 1 y el 0.

**1016 = 1\*16 + 0\*1 = 1610**

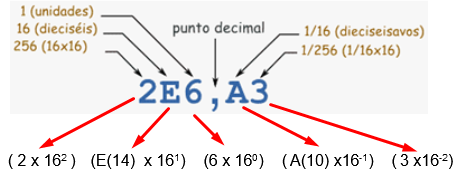
**1B16 = 1\* 16 + 11\*1 = 16 + 11 =2710**

|  |  |  |
| --- | --- | --- |
| **162** | **161** | **160** |
| **256** | **16** | **1** |
|  | 1 | 11 |

**3E16 = 3 · 16 + 14\*1 = 6210**

|  |  |  |
| --- | --- | --- |
| **162** | **161** | **160** |
| **256** | **16** | **1** |
|  | 3 | E |

Operaciones de hexadecimales con decimales



O lo que es lo mismo: 2×16×16 + 14×16 + 6 + 10/16 + 3/(16×16)

La razón para el uso del sistema hexadecimal es que su conversión a binario o la conversión de binario a hexadecimal es muy simple, puesto que, al ser dieciséis igual a dos elevado a cuatro, **cuatro números binarios** componen **un número hexadecimal.**

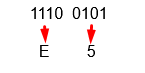
### 3.6.8.- De hexadecimal a binario

El sistema hexadecimal se adoptó en primer lugar debido a lo sencillo que es realizar conversiones ambos. Básicamente, el sistema **hexadecimal** se utiliza como una forma de mostrar la información binaria en una **cadena más corta**. Este gráfico es lo único que necesitarás para convertir un número a otro:

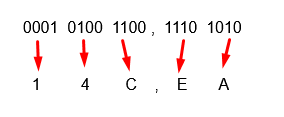
|  |  |
| --- | --- |
| **Hexadecimal** | **Binario** |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

### 3.6.8.- De binario a hexadecimal

cada dígito **hexadecimal** se representa mediante un número **binario de 4 dígitos**. Por ejemplo: convertir el número binario 11100101 a hexadecimal



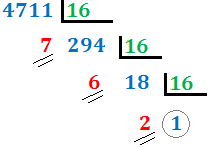
Con decimales sería lo mismo : **0001**0100**1100**,1110**1010**



3.6.9.- Decimal 4711🡪a hexadecimal 1267.

Divisiones sucesivas entre 16 🡪

Se **realizan sucesivas divisiones** entre el **numero de la base destino** y se guardan los restos. Estos restos en orden **inverso** al obtenido son los dígitos del número de la base destino

****

**+ Ejemplos:**

-octal 356🡪 a decimal 192+40+6 = 238

|  |  |  |
| --- | --- | --- |
| 3 | 5 | 6 |
| 3·82=192 | 5·81=40 | 6·80=6 |

-hexadecimal EE🡪 a Decimal 224+14=238

|  |  |
| --- | --- |
| E | E |
| E·161=224 | E·160=14 |