



Desarrollo de Aplicaciones Web

Sistemas informáticos





Tema 1: Introducción a los sistemas informáticos

1. Componentes de un sistema informático. Elementos y conceptos fundamentales

La informática es la rama de la ingeniería que estudia el hardware, las redes de datos y el software necesarios para tratarlos de forma automática. Esta ciencia estudia desde lo relacionado con la arquitectura física de un equipo y su fabricación, hasta los temas referidos a la organización y almacenamiento de información, pasando por todo lo relativo a la creación y uso del software, o a la formación del personal informático. Para ello se basa en múltiples ciencias como las matemáticas, la física, la electrónica, etc.

Sistema informático

Es un conjunto de elementos físicos (hardware) y lógicos (software) interconectados entre sí, destinados a gestionar el tratamiento automático y racional de la información, entendiendo por esto, su organización, su transmisión, su procesamiento y/o su almacenamiento.

Si se habla de un sistema informático, se ha de mencionar al conjunto de personas que lo utiliza, ya sean usuarios, administradores, programadores, etc. Los sistemas informáticos son creados, desarrollados y utilizados por humanos para su propio provecho.

La distinción principal, en un sistema informático, es entre hardware y software:

- Hardware es todo elemento físico que forma parte del ordenador: teclado, ratón, monitor, placa base, procesador, memoria, disco duro, cables, etc. Es la "maquinaria" necesaria para después poder hacer un tratamiento automático de la información.
 - Software es el elemento lógico, es todo aquello que es intangible. Es el conjunto de programas y datos que permiten manejar el hardware, controlando y coordinando su funcionamiento para que realice las tareas deseadas. El software está compuesto por programas y datos:
 - Los programas están formados por un conjunto de órdenes o instrucciones que se utilizan para procesar los datos que se le introducen como información. Son necesarios para la gestión y el control de los equipos y de los trabajos de los usuarios.
 - Los datos son en sí la información que los programas deben procesar, utilizando para ello los diferentes elementos hardware que componen el sistema informático. Son, en definitiva, el objeto o razón de ser del sistema informático.
-



Mucho ha cambiado la cosa desde que en un principio todos los componentes de un sistema informático: físicos, lógicos y humanos; estaban localizados en un mismo lugar. Ahora mismo podemos encontrar sistemas informáticos formados por subsistemas interconectados a través de redes, que pueden llegar a estar a miles de kilómetros entre sí, integrando sistemas complejos de procesamiento de la información. Y estos subsistemas pueden estar compuestos tanto por un superordenador, como por un solo ordenador personal, o por redes locales de ordenadores, o por una combinación de todos ellos.

El sistema informático más simple estará formado por un sólo ordenador y por un usuario que ejecuta los programas instalados en él.





2. Hardware de un sistema informático

Se define ordenador como una máquina electrónica, con algunas partes mecánicas, compuesta por, al menos, una unidad de proceso, y por equipos periféricos, controlada por programas que deben estar almacenados en su memoria central, destinada al tratamiento automático de la información que le es suministrada. Es una máquina de propósito general ya que puede realizar gran variedad de trabajos a gran velocidad y con gran precisión.

La base de todo ordenador es el microprocesador, que es su cerebro y es lo que lo define como tal. El resto de componentes que se le conectan no son más que dispositivos mediante los que se alimenta de energía, o que le permiten interactuar con su entorno, etc.

Existen muchos tipos de ordenadores que pueden clasificarse siguiendo diversos criterios como el tamaño, el propósito para el que se ha creado o tipo.

Vamos a clasificarlos por su tamaño, de mayor a menor, en cuatro tipos:

- Superordenadores o supercomputadores.
- Mainframes o macro computadoras.
- Minicomputadora o miniordenador.
- Ordenadores personales (PC).

2.1. Superordenadores

Un Superordenador es un ordenador extraordinariamente rápido con capacidades de proceso, de cálculo, y de almacenamiento, etc. muy superiores tecnológicamente comparado con el resto de ordenadores contruidos en la misma época.

Físicamente son de gran tamaño. Deben ser instalados en ambientes controlados para poder disipar el calor producido por sus componentes, lo que no impide que puedan soportar la conexión en línea de miles de usuarios. Suelen incorporar numerosos procesadores de gran capacidad trabajando conjuntamente, en paralelo, destinados a una tarea específica.

Su gran número de procesadores les permite procesar ingentes cantidades de información en poco tiempo, pudiendo llegar a procesar miles de millones de operaciones por segundo. Están diseñados para desarrollar cálculos complicados a gran velocidad. Por ello son utilizados para realizar simulaciones de procesos muy complejos con una gran cantidad de datos como, por ejemplo, el análisis del genoma humano, la simulación de explosiones nucleares, las predicciones meteorológicas o astronómicas, etc. Pero también son utilizadas para diseñar y probar virtualmente máquinas complejas como automóviles o aviones, y para controlar el funcionamiento de naves espaciales y satélites, entre otras cosas.

Como también tienen un costo excesivo, en comparación con otros ordenadores, se suelen fabricar muy pocos y bajo pedido.

Tenemos el ejemplo del supercomputador MareNostrum 5 que, previsiblemente, se va a construir en Barcelona con una potencia pico de 314 petaFlops, hasta 17 veces superior a la actual (13,7peta Flops) y 10000 veces mayor que la del superordenador que inicio la saga allá por 2004.



2.2. Mainframe, Macro computadora u Ordenador Central

Los mainframes son grandes ordenadores, de uso general, que disponen de amplio número de procesadores que pueden trabajar de forma independiente entre sí, pudiendo así ejecutar muchas tareas a la vez. Están preparados para realizar varios millones de operaciones por segundo y su gran capacidad de proceso les permite, por un lado, controlar al mismo tiempo a cientos de usuarios y, por otro, almacenar grandes cantidades de información.

En comparación con un superordenador, un mainframe es mucho más barato y puede ejecutar simultáneamente mayor número de programas, pero los superordenadores pueden ejecutar un solo programa mucho más rápido. Son utilizados en las empresas de gran tamaño, con muchas sucursales, como bancos, compañías de transportes, administraciones públicas, etc.

2.3. Minicomputadora o Miniordenador

Son la versión reducida de un mainframe, con menos prestaciones en velocidad, memoria, capacidad de almacenamiento y número de terminales. Están orientadas a dar servicio a empresas e instituciones de menor tamaño, que no necesitan toda la capacidad de proceso, ni todos los periféricos de un mainframe.

Un minicomputador es, por tanto, un sistema multiproceso y multiusuario que ofrece servicios específicos, que cuenta con capacidad para soportar un número relativamente grande de usuarios conectados simultáneamente y que soporta un número limitado de dispositivos. Es de pequeño tamaño, comparado con un mainframe, y de menor coste que éste.

Se suelen utilizar para el almacenamiento de grandes bases de datos, para control automático en la industria y para aplicaciones multiusuario.

2.4. Ordenador Personal o Microordenador

El ordenador personal, conocido como PC, es el ejemplo más claro de un microordenador. Se trata de un equipo de propósito general, de pequeño tamaño y con, al menos, un microprocesador, que suele disponer de ratón y teclado para introducir datos, de un monitor para mostrar la información, y de algún dispositivo de almacenamiento en el que instalar el sistema operativo y guardar datos y programas. Además, admite la conexión de otros periféricos con múltiples y variadas funcionalidades.

Son los ordenadores más accesibles para cualquier tipo de usuario, en cuanto a coste y a facilidad de uso. En sus inicios sólo podían trabajar en modo monousuario, pero que con los avances tecnológicos ahora ya pueden ser utilizados en modo multiusuario e incluso, como servidores de una red de ordenadores.

Los PC's tuvieron su origen gracias a la creación de los microprocesadores por parte de Intel, y a que IBM los incorporó en unos pequeños ordenadores que con el tiempo se estandarizaron, facilitando que otras compañías también pudieran fabricarlos y comercializarlos a precios asequibles al gran público.

La miniaturización ha permitido la creación de otros tipos de microordenadores como los ordenadores portátiles, notebooks, tabletas, PDAs o smartphones.



3. La información y su representación. Sistemas de numeración y cambios de base

Dato es cualquier número, imagen, sonido, símbolo, etc. que no permite tomar ninguna decisión, por no tener un significado concreto. Los datos deben procesarse para dotarlos de significado y convertirlos en información. Eso ocurre cuando trabajamos con un sistema informático, los resultados que muestra el equipo son información que ayuda en alguna tarea.

Los humanos representan la información mediante un conjunto de símbolos (alfabeto) que permiten crear palabras que se asocian a determinados conceptos. Agrupando y combinando estas palabras o conjuntos de símbolos nos expresamos y transmitimos información. Para manejar información numérica utilizamos un sistema de representación compuesto por diez dígitos distintos. Sin embargo, esta simbología que entendemos los humanos no es válida para los ordenadores, ya que éstos son sistemas electrónicos formados por transistores cuyas líneas pueden estar en dos estados distintos: activas o inactivas. Por ello, a pesar de que una línea eléctrica puede conducir un rango continuo de distintos voltajes, los sistemas electrónicos de un computador discretizan ese continuo y sólo consideran dos posibles estados, traducidos en unos y ceros. Esto quiere decir que los ordenadores utilizan sólo dos símbolos ("0" y "1") para representar internamente cualquier tipo de información, ya sean números, textos, imágenes, etc. A este sistema de representación que utiliza dos dígitos se le denomina sistema binario. Dado que los sistemas binarios forman parte de los denominados sistemas de numeración posicionales, vamos a ver algunas cuestiones generales sobre este tipo de sistemas.

Un sistema de numeración en base **b** es una representación de números mediante un alfabeto compuesto por **b** símbolos, dígitos o cifras distintas. En estos sistemas todo número está representado por un conjunto de cifras, contribuyendo cada una de ellas con un valor que depende de: la cifra en sí y de la posición que ocupa la cifra en cuestión dentro del número. Para ver un ejemplo, recurriremos al sistema de numeración más conocido: el de base 10. Es el sistema decimal porque cada cifra puede tomar diez posibles valores:

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

En el ejemplo siguiente, cada cifra del número 15.842 aporta un valor dependiendo de la cifra en sí y de la posición que ocupa la cifra dentro del número:

10.000 5.000 800 40 2 -->15.842

Podemos decir entonces que el número 15.842 puede obtenerse como la suma de las cantidades 10.000, 5.000, 800, 40 y 2. Esto es:

$$\begin{aligned} 15.842 &= 10.000 + 5.000 + 800 + 40 + 2 \\ &= 1 \times 10.000 + 5 \times 1.000 + 8 \times 100 + 4 \times 10 + 2 \times 1 \\ &= 1 \times 10^4 + 5 \times 10^3 + 8 \times 10^2 + 4 \times 10^1 + 2 \times 10^0 \end{aligned}$$

Como vemos en la descomposición anterior, cada cifra del número tiene un peso específico que se va multiplicando por la base, en este caso 10, cada vez que nos desplazamos hacia la izquierda. Si generalizamos este funcionamiento, podríamos decir que:

En un sistema de numeración en base $b > 1$, todo entero N positivo tiene una única representación de la forma:

$$N = n_p b^p + n_{p-1} b^{p-1} + \dots + n_1 b^1 + n_0 b^0$$

Donde $0 \leq n_i < b$ para todo $i = 0, 1, \dots, p$.

3.1. Sistema de numeración binario

En el sistema de numeración binario la base es 2 ($b = 2$), por lo que se utiliza un alfabeto de tan sólo dos elementos $\{0, 1\}$ para representar cualquier número. Los elementos de este alfabeto se denominan cifras binarias o bits. Por ejemplo, los 8 primeros números decimales en binario serían:

00	01	10	11	100	101	110	111
0	1	2	3	4	5	6	7

Si nos fijamos en el ejemplo, vemos que el número de bits utilizados en la representación determina el número de valores distintos que podemos representar:

- Con 2 bits podemos representar 4 valores distintos ($4 = 2^2$)
- Con 3 bits podemos representar 8 valores distintos ($8 = 2^3$)

Generalizando podemos decir que si utilizamos n bits, podremos representar 2^n valores distintos, que van desde el 0 al $2^n - 1$.

3.2. Convertir de binario a decimal

Para convertir un número binario a un número decimal, basta con aplicar la fórmula general presentada para la representación de números en un sistema posicional, pero en este caso, teniendo en cuenta que la base es 2. Como ejemplo se va a transformar el número binario de seis dígitos 110101 a decimal:

$$\begin{aligned} N &= n_p b^p + n_{p-1} b^{p-1} + \dots + n_1 b^1 + n_0 b^0 \\ 110101 &= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\ &= 32 + 16 + 4 + 1 = 53 \end{aligned}$$

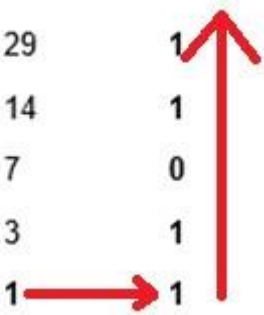
Si nos fijamos en el ejemplo, transformar de binario a decimal es bastante sencillo, pues basta con sumar los pesos en cuyas posiciones haya un 1.

3.3. Convertir de decimal a binario

Para convertir un número entero decimal a binario hay que dividir el número de partida entre la base 2 (sin obtener decimales en el cociente) y seguir dividiendo entre 2 los cocientes que se vayan obteniendo sucesivamente. Los restos de estas divisiones y el último cociente (que serán siempre menor que la base, esto es, 1 o 0) colocados en orden inverso al que han sido obtenidos determinarán la representación binaria del número decimal original.

Como ejemplo vamos a ver cómo convertir el número decimal 59 a binario:

División	Cociente	Resto
59 / 2	29	1
29/2	14	1
14/2	7	0
7/2	3	1
3/2	1	1



La representación binaria del número decimal 59 es **111011**, cogiendo el último cociente primero y todos los restos en orden del último al primero.

$$\begin{aligned} 111011 &= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 \\ &= 32 + 16 + 8 + 2 + 1 = 59 \end{aligned}$$

EJERCICIO 1

- Realiza las siguientes conversiones de decimal a binario:
 - 340
 - 176
 - 9
 - 521
 - 1480
 - 75
- Haz la conversión inversa para comprobar que, efectivamente, las conversiones primeras han sido correctas.



3.4. Sistemas octal y hexadecimal

Al principio se utilizaba solo el sistema binario para interpretar y transformar los datos, con lo que programar era algo realmente tedioso. Se recurrió entonces al uso de sistemas intermedios para traducir desde y hacia el sistema binario. Son el octal y el hexadecimal.

El sistema octal tiene como base de numeración el 8 (símbolos del 0 al 7) y también se trata de un sistema posicional (recordamos que esto implica que un mismo dígito tiene distinto valor dependiendo de la posición que ocupa).

El sistema hexadecimal tiene como base 16 y, al igual que los anteriores, también se trata de un sistema posicional.

3.5. El sistema hexadecimal

El sistema hexadecimal tiene una base mayor a 10, por lo que necesitamos más de 10 signos. En este sistema de numeración tenemos los signos del 0 al 9 y se completa con 6 nuevos que en este caso son letras, y que su equivalencia en decimal sería la siguiente:

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

Las conversiones de hexadecimal a decimal y viceversa siguen los mismos algoritmos que utilizamos para los sistemas de numeración de bases inferiores a 10.

Veamos el ejemplo de pasar 1234 en base 10 a hexadecimal:

División	Cociente	Resto
1234/16	77	2
77/16	4	13

Entonces tenemos que la posición 0 será 2, la posición 1 será D(13) y la posición 3 será 4.

Por lo tanto, $1234 = 4D2$.

Y a la inversa, hacemos el desarrollo de las potencias:

$$45AB = 4 \times 16^3 + 5 \times 16^2 + A(10) \times 16^1 + B(11) \times 16^0 = 17835$$

Como vemos, una cifra en hexadecimal ocupa menos posiciones, necesita menos dígitos, que en decimal. Y si comparamos con el binario, aun es mayor la diferencia.

$$A5 = 10 \times 16 + 5 = 165 = 10100101$$



En este caso, en binario ocupamos justamente 4 veces más de posiciones ($16 = 2^4$) que en hexadecimal. Además, observamos que el resultado se obtendría también cambiando cada cifra a binario, pero reservando para cada una cuatro posiciones binarias:

$$\begin{array}{l} A=1010 \quad 5=0101 \\ \text{Por lo tanto} \\ A5=1010 \ 0101 \end{array}$$

Esta es una propiedad fundamental del hexadecimal: su gran facilidad para la traducción a binario (y viceversa) y su mayor compacidad (más compacto). Este es el motivo por el que se trabaja en informática con hexadecimal.

Podemos ver su uso abriendo el código fuente de cualquier página y buscando alguna etiqueta de color. Ahí están definidos con el sistema rgb en hexadecimal.

3.6. Resumen

$$N = n_p b^p + n_{p-1} b^{p-1} + \dots + n_1 b^1 + n_0 b^0$$

Para pasar de cualquier sistema de numeración de los que hemos visto a decimal:

Donde $0 \leq n_i < b$ para todo $i = 0, 1, \dots, p$. Siendo:

- n el carácter que toca tratar
- p la posición que ocupa
- b la base del sistema de numeración desde el que se hace la conversión.

Para pasar de decimal a binario, octal o hexadecimal, divisiones donde:

- el dividendo es el valor que queremos convertir.
- el divisor es la base a la que vamos a convertir el dato.
- el resto y el último cociente formarán el nuevo valor, ordenándolos desde el último cociente hasta el primer resto.

Para las conversiones entre binario y hexadecimales:

- Partimos de la tabla de equivalencias entre hexadecimal y binario.
- De Hexadecimal a binario:
 - Reservamos para cada carácter en hexadecimal 4 posiciones binarias que serán ocupadas por su equivalente en binario. Se completa con ceros a la izquierda si hay menos de 4 dígitos binarios.
- De binario a hexadecimal:
 - Agrupamos los caracteres binarios de 4 en 4 comenzando por el final. El último que nos quede se completa con ceros a la izquierda, en caso de tener menos de 4 caracteres. Hacemos la conversión de cada cuarteto, cambiando los dígitos mayores de 9 a su equivalente en el sistema



hexadecimal.

EJERCICIO 2

- Resuelve los diferentes ejercicios propuestos en los siguientes enlaces:

<https://www.matesfacil.com/ESO/sistemas-numeracion/base-octal/sistema-numeracion-octal-base-ocho-ejemplos-teoria-propiedades-cambio-base-decimal-ejercicios-resueltos.html>

<https://www.matesfacil.com/ESO/sistemas-numeracion/base-hexadecimal/sistema-numeracion-hexadecimal-base-16-ejemplos-teoria-propiedades-cambio-base-decimal-ejercicios-resueltos.html>



4. Operaciones con binarios

Como ya sabemos, el sistema binario está compuesto por dos dígitos, el 0 y el 1. Y, ahora que ya conocemos este sistema en base 2, vamos a ver cómo realizar operaciones sencillas de sumas y restas en binario.

4.1. Suma de números binarios

Las reglas básicas a seguir para la operación de suma de números binarios son:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

El resultado de esa 4ª regla implica que, cuando se sume $1 + 1$, el resultado sea 0 y haya que “llevarse” 1.

Vamos a verlo con un ejemplo:

$$00111 + 11010 = 100001$$

1 1 1

00111

11010

100001

- La operación se realizará siempre de derecha a izquierda.
- $1 + 0 = 1$ colocamos el 1 y no hay llevadas.
- La siguiente es $1 + 1 = 10$. El 0 se deja en el resultado final y el 1 será la llevada (en rojo).
- La tercera operación implica $1 + 0 = 1$, y a ese resultado de 1 se le suma la llevada, por lo que $1 + 1 = 10$ y, de nuevo, el 0 se deja en el resultado final y el 1 será la llevada (en rojo).
- En la cuarta columna se repite la misma casuística de la tercera. Tras realizar las operaciones necesarias se deja el 0 se deja en el resultado final y el 1 será la llevada (en rojo).
- Y por último, $1 + 0 = 1$, a éste 1 le sumamos el 1 de la llevada volviendo a tener como resultado 10, pero, como en este caso no hay más operaciones que realizar, se deja el 10 en el resultado final.

4.2. Resta de números binarios

En este caso, las reglas básicas a seguir son:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 11$$



El resultado de esa 4ª regla implica que, cuando sea 0 - 1, el resultado sea 1 y haya que “llevarse” 1.

Vamos a verlo con un ejemplo:

1101 + 0011 = 1010	
	¹
	1101
	0011
<hr/>	
	1010

- La operación, al igual que para la suma, se realizará siempre de derecha a izquierda.
- 1 - 1 = 0 colocamos el 0 y no hay llevadas.
- La siguiente es 0 - 1 = 11. Un 1 se deja en el resultado final y el otro será la llevada (en rojo).
- La tercera operación implica 1(el rojo) - 1 = 0 y, a continuación, 0 - 0 que es 0.
- Y, por último, 1 - 0 = 1.

EJERCICIO 3

- Resuelve las siguientes sumas entre números binarios:
 - 101101 + 11
 - 1110101 + 001011110
 - 1101 + 1001
 - 10101 + 100001
 - 01 + 111
 - 1011 + 11111
- Resuelve, ahora, las siguientes restas de números binarios:
 - 11101 - 110
 - 1110101 - 1011110
 - 1101 - 1001
 - 100001 - 10101
 - 111 - 01
 - 11111 - 1011
- En el Aula Virtual de la asignatura tienes el documento “Sumas y Restas con Números Binarios” con más ejercicios.

4.3. Multiplicación y división de números binarios

Estas dos operaciones no difieren en nada del procedimiento seguido para multiplicar o dividir números decimales.

5. Operaciones lógicas

La lógica binaria es la que trabaja con variables binarias y operaciones lógicas. Así las variables solo toman dos valores discretos: verdadero y falso; aunque estos dos valores lógicos también se pueden denotar como si y no o como 1 y 0 respectivamente.

Esta es la base de los sistemas digitales y ello implica también la base de la estructura de los computadores.

En la lógica binaria representamos los dígitos utilizando exclusivamente los valores 0 y 1, números que no tienen un valor numérico real, sino que más bien de tipo discreto, es decir, representan los distintos estados del objeto de estudio, por ejemplo, en la tarea de poder desarrollar un circuito digital.

Generalmente, el estado lógico 0 representa ausencia de tensión y el estado lógico 1 representa existencia de tensión. Mediante la combinación de estos valores es posible generar una serie de datos convertibles a cualquier código utilizando la normativa aplicable en cada caso.

A continuación, se van a ver las tablas de verdad de las operaciones binarias fundamentales. Una tabla de verdad no es más que los valores que pueden resultar de una operación lógica concreta teniendo en cuenta todas las combinaciones posibles.

Negación lógica

Es una función unaria que invierte el valor lógico de su argumento, por lo que también se conoce como **NOT**. Suele simbolizarse por una barra horizontal sobre su argumento:

P1	R
1	0
0	1

Unión o suma lógica

Es una función de varios argumentos que vale 0 solo si todos sus argumentos valen 0. En el resto de casos vale 1. Se denomina también función **OR**.

P1	P2	R
0	0	0
0	1	1
1	0	1
1	1	1

Intersección, producto o multiplicación lógicos

Es una función de varios argumentos que vale 1 solo si todos sus argumentos valen 1. Se denomina función **AND**.


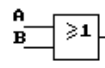
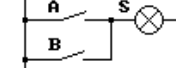

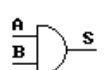
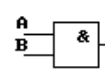
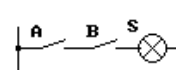
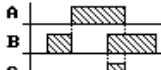
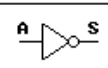
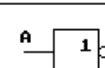
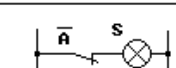
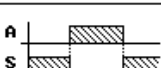

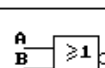
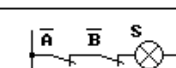
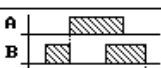
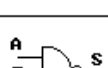
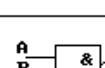
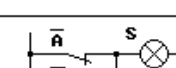

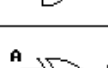
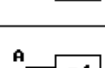
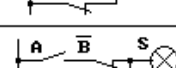

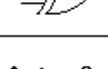
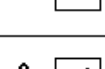
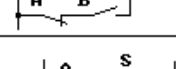

P1	P2	R
0	0	0
0	1	0
1	0	0
1	1	1

Unión exclusiva

También llamada OR exclusiva, es una función de varios argumentos que vale 1 solo si uno, y solo uno, de sus argumentos vale 1. En el resto de casos vale 0.

P1	P2	R
0	0	0
0	1	1
1	0	1
1	1	0

Resumen

Función	Ecuación lógica	Símbolos			Tabla de verdad	Cronograma															
		Norma MIL	Norma IEC	Circuito físico con contactos																	
OR	$S = A + B$				<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	S																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	1																			
AND	$S = A \cdot B$				<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1	
A	B	S																			
0	0	0																			
0	1	0																			
1	0	0																			
1	1	1																			
NOT	$S = \bar{A}$	 inversor			<table><tr><td>A</td><td>S</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	S	0	1	1	0										
A	S																				
0	1																				
1	0																				
NOR (OR+NOT)	$S = \overline{A + B}$ $S = \bar{A} \cdot \bar{B}$				<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0	
A	B	S																			
0	0	1																			
0	1	0																			
1	0	0																			
1	1	0																			
NAND (AND+NOT)	$S = \overline{A \cdot B}$ $S = \bar{A} + \bar{B}$				<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0	
A	B	S																			
0	0	1																			
0	1	1																			
1	0	1																			
1	1	0																			
EXOR	$S = A \oplus B =$ $= A \bar{B} + \bar{A} B$				<table><tr><td>A</td><td>B</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0	
A	B	S																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	0																			
EXCI TADOR	$S = A$				<table><tr><td>A</td><td>S</td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	S	0	0	1	1										
A	S																				
0	0																				
1	1																				



EJERCICIO 4

- Resuelve los ejercicios del documento “Ejercicio Puertas lógicas” que encontrarás en el Aula Virtual.





6. Licencias del software

Comenzaremos definiendo algunos conceptos clave para entender gran parte de lo que rodea a las licencias software.

En primer lugar, las licencias software nos sirven para establecer un contrato entre el autor de una aplicación software (sometido a propiedad intelectual y a derechos de autor) y el usuario. En el contrato se definen con precisión los derechos y deberes de ambas partes, es decir, los “actos de explotación legales”.

Por otra parte, entendemos por derecho de autor o copyright la forma de protección proporcionada por las leyes vigentes en la mayoría de los países para los autores de obras originales incluyendo obras literarias, dramáticas, musicales, artísticas e intelectuales, tanto publicadas como pendientes de publicar.

Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el usuario. Distinguimos varios tipos de software o licencias en función de lo limitadas que estén las acciones del usuario sobre el mismo:

- **Software propietario:** Es aquel software cuya copia, redistribución o modificación están prohibidos por su propietario, que tendrá que dar permiso (o venderlo de alguna manera) para que su software pueda ser copiado, modificado, etc. Es el llamado software con copyright.
- **Software libre:** licencia de software que permite que los usuarios puedan ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Tiende hacia el copyleft, es decir, cualquier programa derivado debe seguir siendo siempre libre. Se basa en las 4 libertades esenciales del usuario definidas por la Free Software Foundation.
- **Software Open Source:** código diseñado de manera que sea accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente. Algunas licencias open source incluso permiten privatizar el software derivado. Cualquier desarrollo que se quiera distribuir como open source, debe cumplir los 10 requisitos definidos por la Open Source Initiative.
- **Software semilibre:** software no libre con autorización de uso, copia, modificación y redistribución sin ánimo de lucro.
- **Software de dominio público:** software libre que no tiene un propietario por lo que no hay derechos de autor, licencias o restricciones de distribución y el software puede ser libremente utilizado.
- **Software con copyleft:** licencia que permite la libre distribución de copias y versiones modificadas de un trabajo u obra, reclamando que los mismos derechos se respeten en las distintas versiones que se cambien.

EJERCICIO 4:

- Busca aplicaciones de ejemplo de cada licencia vista.
-



7. Requisitos del software. Instalación y prueba. Comparación

Software de un sistema informático

Definimos el concepto de sistema informático como un conjunto de elementos que hacen posible el tratamiento automatizado de la información. Vamos a centrarnos en el software de un sistema informático. Éste está formado por programas, estructuras de datos y documentación asociada. Así, el software está distribuido en el ordenador, los periféricos y el subsistema de comunicaciones. Ejemplos de software son los sistemas operativos, paquetes ofimáticos, compresores, editores de imágenes y un sinfín de programas más o menos específicos según el conocimiento y ámbito profesional del usuario.

7.1. Requisitos e instalación: Determinación del equipo necesario

¿Qué ocurre cuando queremos instalar una aplicación software en el ordenador? En todo proceso de instalación se han de seguir unos pasos que describiremos a continuación. Si no se realizan adecuadamente podemos encontrarnos con un funcionamiento limitado o erróneo de la aplicación. Los pasos serían:

- Determinación del equipo necesario.
- Ejecución del programa de instalación.
- Configuración de la aplicación.

Determinación del equipo necesario Lo primero que debemos hacer es conocer qué necesita la aplicación para que funcione adecuadamente en el ordenador, es decir, qué características o requisitos necesarios tendrá que tener el sistema informático. Cada desarrollador crea sus aplicaciones enfocadas a plataformas concretas, con unas necesidades de hardware y software necesarias para su funcionamiento. Una aplicación creada para una plataforma no podrá ser instalada en otra distinta. Tampoco podrá ser instalada la aplicación si nuestro sistema informático no cumple los requisitos mínimos. Antes de proceder a la instalación de una aplicación tendrá que reunir la información sobre el hardware de su ordenador y deberá verificar que su hardware le permite realizar el tipo de instalación que desea efectuar.

Las características para que la aplicación se ejecute adecuadamente pueden ser de naturaleza hardware:

- Plataforma hardware: PC, Mac, etc.
 - Procesador: fabricante, velocidad, generalmente se indica el inferior posible de lagama con el que la aplicación funciona adecuadamente.
 - Memoria RAM mínima.
 - Espacio mínimo disponible en el soporte de almacenamiento: por ejemplo, endisco duro o unidad de almacenamiento externa para aplicaciones portables.
 - Tarjeta gráfica: la memoria gráfica necesaria para el buen funcionamiento de la aplicación.
 - Resolución recomendada del monitor.
-



Y de carácter software:

- Plataforma software: sistema operativo bajo el que funciona la aplicación, Windows, Linux, etc.
- Otros paquetes software adicionales necesarios, tales como actualizaciones concretas de seguridad para el sistema operativo, la JVM (máquina virtual de Java), el Flash Player, etc. Por ejemplo, para instalar el editor de imágenes de Microsoft te indica que debes tener instalado varios componentes de Microsoft.

Teniendo en cuenta lo visto anteriormente, los fabricantes de aplicaciones informáticas suelen establecer tres niveles de requisitos para la instalación de sus aplicaciones:

- Equipo básico.
- Equipo opcional.
- Equipo en red.

7.2. Requisitos e instalación: Ejecución del programa de instalación

Por la instalación de un programa o aplicación informática entendemos el conjunto de pasos que nos van a permitir copiar los archivos necesarios, configurar, implantar y poner en funcionamiento una aplicación en un sistema informático.

La mayoría de las aplicaciones presentan dos niveles en función de los conocimientos del usuario:

- Instalación básica
- Instalación personalizada o avanzada

Instalación básica

Este nivel está diseñado para usuarios con pocos conocimientos informáticos. El programa realizará una instalación en función de los elementos que detecte en el equipo y según unos parámetros básicos establecidos por defecto por el fabricante.

Instalación personalizada o avanzada

Permite al usuario experto incluir o eliminar elementos de la aplicación con el fin de optimizar los recursos sistema informático, instalando sólo aquellos elementos de la aplicación que se van a utilizar. Por ejemplo, la instalación personalizada del paquete Microsoft Office permite elegir los programas a instalar (Microsoft Word, Excel, PowerPoint, Frontpage, etc.).

Cuando se adquiere una aplicación informática la aplicación se encuentra normalmente en formato comprimido. El traspaso del programa al soporte de almacenamiento de nuestro ordenador, normalmente el disco duro, se realiza a través del programa de instalación (su nombre puede ser setup, install, instalar, etc.), y es el encargado de extraer los bloques de la aplicación de los discos, descomprimiéndolos si es necesario; crear la estructura de directorios necesaria, ubicar los archivos de la aplicación donde corresponda, y, si fuera necesario, modificar el registro del sistema.



7.3. Requisitos e instalación: Configuración de la aplicación

Una vez realizado correctamente el proceso de instalación sólo queda configurar las opciones de la aplicación, a veces también del sistema operativo, y configurar el entorno de trabajo. En la configuración se pueden modificar los parámetros establecidos por defecto para la aplicación. Algunas aplicaciones pueden generar una serie de archivos de configuración con los datos introducidos por los usuarios. El usuario debe realizar una última tarea antes de comenzar a utilizar la aplicación: configurar el entorno de trabajo. Esto consiste en definir una serie de parámetros de funcionamiento que adecuen el funcionamiento de la aplicación a las exigencias del usuario. Este proceso se realizará en el caso que no sea satisfactoria la configuración establecida por defecto por el programa. Entre estos parámetros aparecen el ajuste y distribución de la pantalla, definición de directorios de trabajo, etc.

