UD 1 – SISTEMAS INFORMÁTICOS

ÍNDICE

1. INTRODUCCIÓN

Se puede decir que la **información** es un conjunto de datos, ordenados adecuadamente, que aportan conocimiento sobre alguna cosa.

El término **Informática** ha evolucionado a lo largo del tiempo. Al principio, se definió como la ciencia que se encargaba de estudiar el tratamiento automático de la información. Procede de la concentración de dos palabras: **infor**mación y auto**mática**.

Hay varias **razones** por las que puede ser necesario **automatizar** el tratamiento de la información, por ejemplo agilizar operaciones que el ser humano tardaría mucho tiempo en realizar, realizar operaciones monótonas, realizar tareas que el ser humano no podría hacer solo, etc.

Algunas de las **operaciones**, de forma genérica, que se suelen realizar sobre los datos son: agrupación, ordenación, cálculos y reducción.

Algunas de las **cualidades** que ha de tener la información para que sea útil son las siguientes:

- Oportunidad: la información ha de llegar en el momento oportuno para que sea útil.
- La velocidad: cuando más urgente se necesita la información más cara resulta.
- Precisión: la precisión significa dar más detalles.
- Exactitud: la exactitud se mide en términos de porcentaje de error.
- Integridad: la información tiene que ser completa.
- Presentación: forma de ofrecer la información al usuario. Tiene que ser clara y relevante.
- Seguridad: la información tiene que estar debidamente protegida contra el deterioro o el acceso no autorizado.

La Organización Internacional de Estandarización (ISO) define **Sistema Informático** como: *"una o más computadoras, el software asociado, los periféricos, los terminales, los operadores humanos, los procesos físicos, los medios de transmisión de la información, etc., que constituyen un todo autónomo capaz de realizar un tratamiento de la información"*.

La Organización Internacional de Estandarización (ISO), es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización (ON's) nacionales, que produce normas internacionales industriales y comerciales. Dichas normas se conocen como **normas ISO** y su finalidad es la coordinación de las normas nacionales, en consonancia con el Acta Final de la Organización Mundial del Comercio, con el propósito de facilitar el comercio, facilitar el intercambio de información y contribuir con unos estándares comunes para el desarrollo y transferencia de tecnologías.

2. COMPONENTES LÓGICOS DE LOS SISTEMAS INFORMÁTICOS

2.1. Sistemas de codificación

La forma en que los datos se representan en la memoria del ordenador se denomina sistema de codificación, existen muchos sistemas de codificación, nosotros trataremos algunos de ellos.

Antes de estudiar la forma de codificar la información, analizaremos el sistema de numeración **decimal**. Este es el que habitualmente utilizamos en la vida diaria, su base es 10 ya que dispone de 10 símbolos: 0,1,2,3,4,5,6,7,8, y .9. Es importante resaltar que cada uno de los dígitos de un número escrito en un determinado sistema de numeración tiene un valor diferente, dependiendo de la posición que ocupe.

Por ejemplo: el número 647, escrito en base 10, se puede descomponer en:

$$6 \times 10^2 + 4 \times 10^1 + 7 \times 10^0.$$

En este ejemplo, hemos visto la forma de pasar un número escrito en un sistema de numeración cualquiera al sistema de numeración de base 10. Cada dígito se multiplica por la base del sistema de numeración en que está escrito el número elevado al peso del dígito. El peso va aumentando hacia la izquierda en una unidad, empezando por el 0, según los dígitos del número que pretendemos traducir.

El problema de pasar un número en base 10 a otro sistema de numeración se resuelve de la siguiente manera: el número en base 10 y los sucesivos cocientes se dividen por la base del sistema de numeración al que queremos pasar el número, hasta encontrar un cociente menor que la base. El resultado final es el último cociente y los restos en sentido inverso a como se obtuvieron.

Sistema binario (base 2)

Internamente, los ordenadores trabajan con señales eléctricas que representan dos estados: o pasa corriente o no pasa corriente. Por tanto, en un sistema informático la información se representa como combinación de estos dos estados 0 y 1, o dicho de otra forma, se representa en **sistema binario**.

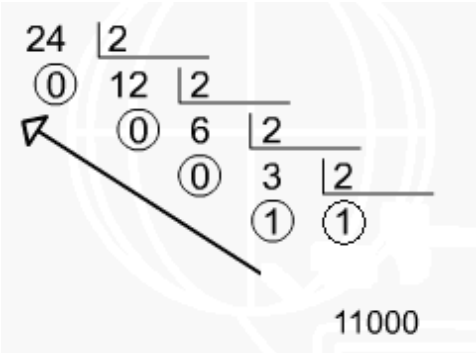
El sistema binario tiene **base 2**, sólo dispone de dos dígitos, 0 y 1. A continuación, vamos a pasar números del sistema binario al sistema decimal y viceversa.

Conversión de binario a decimal: Para convertir un número de binario a decimal se comienza por el lado derecho del numero binario, cada numero se multiplica por la base, que es 2 en binario, elevado a la posición del número, empezando por 0. Después de realizar cada una de las multiplicaciones, se suman todas y él numero resultante será el equivalente al sistema decimal.

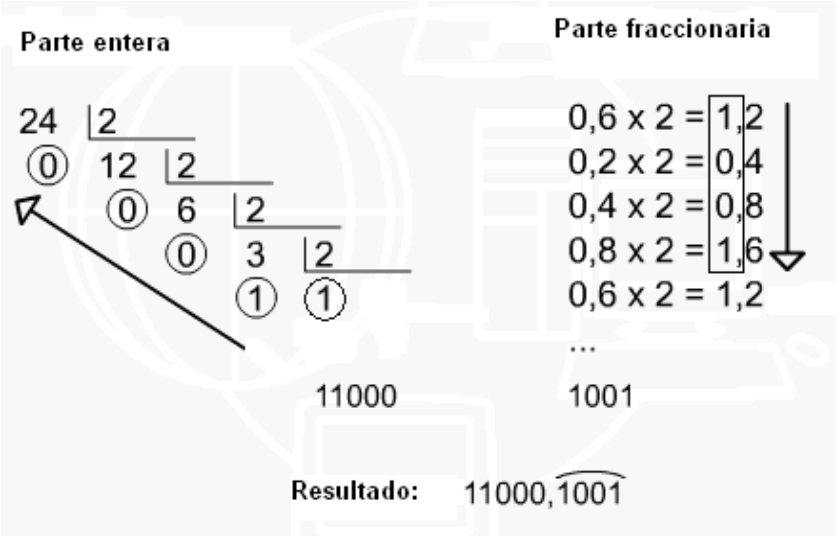
Ejemplo: $11010 = 0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 = 26$

Para realizar esta conversión de manera más rápida, es aconsejable aprender la **tabla de las potencias de 2**: $2^0 = 1$; $2^1 = 2$; $2^2 = 4$; $2^3 = 8$; $2^4 = 16$; $2^5 = 32$; $2^6 = 64$; $2^7 = 128$; $2^8 = 256$; $2^9 = 512$; $2^{10} = 1024$.

Conversión de decimal a binario: Para convertir un número decimal a binario, se divide sucesivamente por 2, y se toman sucesivamente el último cociente y desde el último resto hasta el primero. Como ejemplo vemos como pasar el número 24 a binario.



¿Qué sucede si tenemos parte fraccionaria? La parte fraccionaria se obtiene mediante multiplicaciones sucesivas por la base, quedándonos con la parte entera de la multiplicación. Como ejemplo veremos como pasar a binario el número 24,6.



Sistema hexadecimal (base 16)

La base hexadecimal surgió para compactar la información binaria. Se utiliza un dígito hexadecimal para representar una cadena de 4 dígitos binarios, ya que $16 = 2^4$. Teniendo en cuenta que con 4 dígitos binarios podemos representar 16 números diferentes en el sistema hexadecimal es necesario un alfabeto de 16 dígitos diferentes. Tenemos entonces que los dígitos hexadecimales son: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E y F. A equivale a 10 en base 10. B equivale a 11 en base 10. C equivale a 12 en base 10. D equivale a 13 en base 10. E equivale a 14 en base 10. F equivale a 15 en base 10.

Conversión de binario a hexadecimal: para pasar de binario a hexadecimal solo hay que agrupar los dígitos binarios de cuatro en cuatro, ya que $16 = 2^4$.

Ejemplo:

$$1011100110 = \underbrace{0010}_{2}\ \underbrace{1110}_{E}\ \underbrace{0110}_{6} = 2E6_{(16}$$

Conversión de hexadecimal a binario: para pasar de hexadecimal a binario se sigue el procedimiento inverso al anterior. Cada dígito hexadecimal se transforma en los cuatro dígitos binarios equivalentes a su valor.

Conversión de decimal a hexadecimal y viceversa: para pasar entre estos dos sistemas de codificación se puede utilizar el mismo método explicado para el sistema binario, teniendo en cuenta que en este caso la base a utilizar para realizar las multiplicaciones o divisiones es 16 en lugar de 2. Otra forma de realizar las conversiones es utilizar el paso intermedio de transformar el número al sistema binario y realizar las agrupaciones de cuatro en cuatro.

Sistema octal (base 8)

Al igual que la base hexadecimal, se utiliza para compactar información binaria, pero en este caso, la compactación es menor. Mientras que en la base hexadecimal con un sólo dígito se puede representar una cadena de 4 dígitos binarios, en la base octal un dígito sólo puede representar 3 dígitos binarios, ya que $8 = 2^3$. Los dígitos posibles para la base octal, evidentemente, son los que van del 0 al 7.

Conversión de binario a octal: para pasar de binario a octal solo hay que agrupar los dígitos binarios de tres en tres, ya que $8 = 2^3$.

Ejemplo:

$$1011100110 = \underbrace{001}_{1}\ \underbrace{011}_{3}\ \underbrace{100}_{4}\ \underbrace{110}_{6} = 1346_{(8}$$

Conversión de octal a binario: para pasar de octal a binario se sigue el procedimiento inverso al anterior. Cada dígito octal se transforma en los tres dígitos binarios equivalentes a su valor.

Conversión de decimal a octal y viceversa: es equivalente al método explicado en el sistema hexadecimal, teniendo en cuenta que en este caso la base es 8.

Conversión de hexadecimal a octal y viceversa: estas conversiones no son posibles en una forma directa. Para realizar cualquiera de ellas se deberá pasar a otra base como paso intermedio, lo más sencillo es realizar la conversión intermedia al sistema binario.

En la siguiente tabla se resumen las equivalencias entre los sistemas de codificación vistos:

Decimal	Binari	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

2.2. Aritmética binaria

La Unidad Aritmético Lógica, en la CPU del procesador, es capaz de realizar operaciones aritméticas, con datos numéricos expresados en el sistema binario. Naturalmente, esas operaciones incluyen la adición, la sustracción, el producto y la división.

La suma binaria: la tabla de sumar en binario es la siguiente:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 y me llevo 1, ya que el 2 en binario es 10.

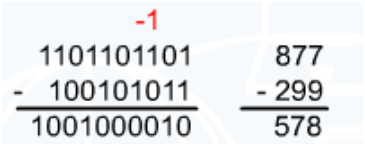
Ejemplo de una suma en binario:

```
 11 1  1
  11101011    235
+ 10110010  + 178
----------  -----
 110011101    413
```

La resta binaria: la tabla de restar en binario es la siguiente:

0 – 0 = 0
1 – 0 = 1
1 – 1 = 0
0 – 1 = 1 y debo restar 1 en el paso siguiente.

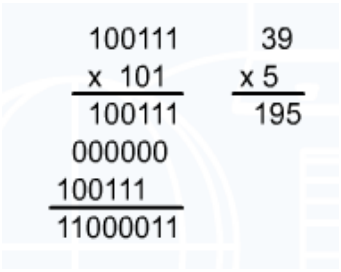
Ejemplo de una resta en binario:



La multiplicación binaria: multiplicar en binario es similar a multiplicar en decimal. Se basa en la siguiente tabla:

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

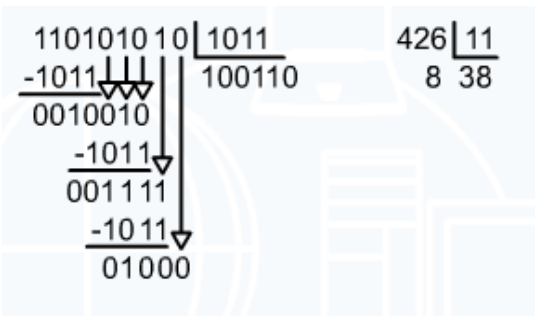
Ejemplo de una multiplicación en binario:



La división binaria: dividir en binario es similar a dividir en decimal. Se basa en la siguiente tabla:

0 : 0 = ?
0 : 1 = 0
1 : 0 = ∞
1 : 1 = 1

Ejemplo de una división en binario:



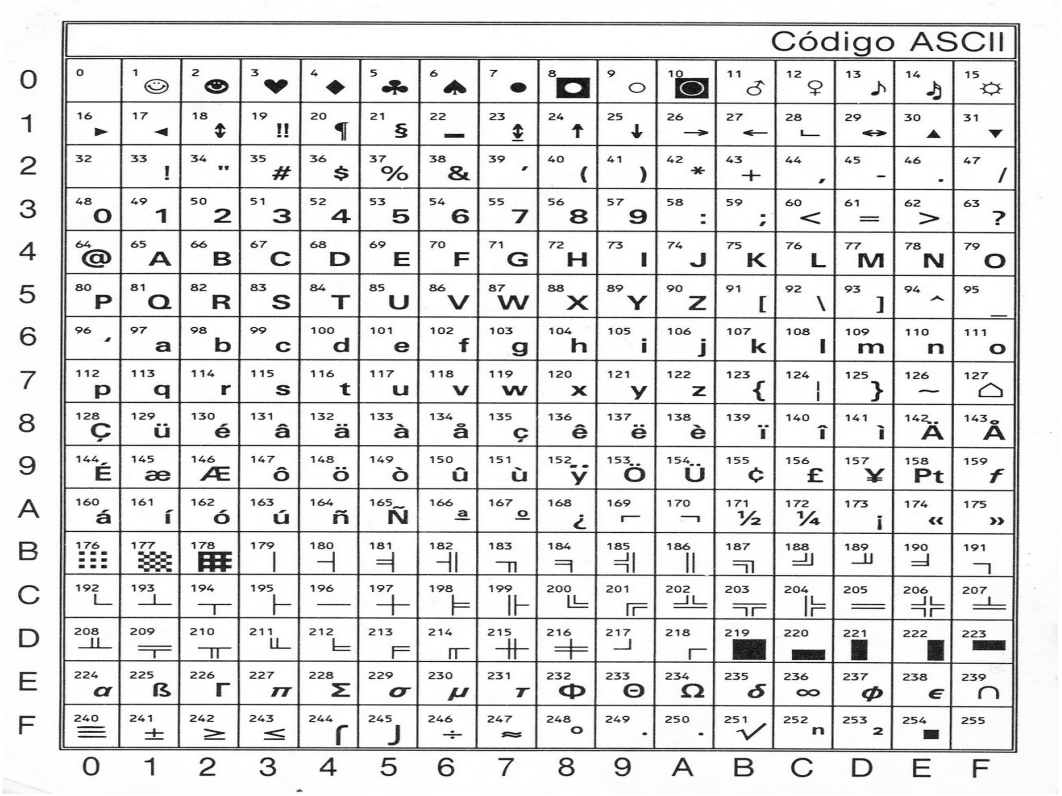
2.3. Representación de la información alfanumérica

La información alfanumérica contiene caracteres alfabéticos, signos especiales (signos de puntuación, paréntesis, etc.) y caracteres numéricos. La representación de todos estos símbolos se realiza asignando a cada uno de ellos una única combinación de unos y ceros.

Existe una tabla de correspondencia que asigna a cada carácter una combinación binaria. Dicha tabla recibe el nombre de código. Existen varios códigos de representación de caracteres normalizados, es decir, que se pretende que todos los ordenadores presenten cada carácter con la misma combinación binaria.

Los sistemas de **codificación alfanumérica** más importantes son:

El código **ASCII** (Código Estándar Americano para el Intercambio de Información) originalmente utilizaba 7 bits, es decir, permitía representar 2^7 = 128 caracteres. Posteriormente se amplió a **8 bits** para poder representar los caracteres especiales de cada idioma, conservando como estándar los caracteres del 0 a 127. A continuación se muestra la tabla ASCII de 8 bits:



El código **EBCDIC** (Clave Extendida de Intercambio Decimal Cifrada en Binario), utiliza **8 bits** con lo cual pueden representar 256 combinaciones distintas. Este código tiende a desaparecer.

El código **FIELDATA,** código que utiliza **6 bits** por símbolo, su implantación está limitada a ordenadores que procesan bloques de 36 bits.

El código **UNICODE,** es un código estándar internacional utilizado en los sistemas operativos. Utiliza **16 bits** por símbolo, lo que permite representar $2^{16} = 65536$ caracteres. Con 16 bits ya se pueden representar todos los caracteres internacionales, pero surge otro problema ¿cómo se almacena la información en el ordenador?, en **low-endian** o en **big-endian,** en palabras simples es la forma en que el procesador de un computador lee la información; de izquierda a derecha o de derecha a izquierda. Las arquitecturas x86 de Intel y los procesadores Alpha de DEC usan low-endian, mientras que las plataformas Sun's SPARC, Motorola, e IBM PowerPC utilizan la convención big-endian.

El problema que se suscitó es que se tenían dos formas de almacenar Unicode, para solucionar ese problema se definió una muy extraña convención: se le agregaría un *FE FF* al inicio de cada Unicode. A esto se le llamó **Marca de Orden Unicode** para big-endian se intercambiaba el orden a *FF FE*.

Actualmente el estándar **UTF-8**, dentro de los sistemas de codificación de UNICODE recoge la codificación de la mayor parte de los signos de escritura de los lenguajes occidentales y también de los orientales, e incluso de símbolos matemáticos. Para ello UTF-8 utiliza **entre uno y seis bytes** para describir cada signo:

- El bit más significativo de un carácter de byte-simple es siempre 0
- Los bits más significativos del primer byte de una secuencia multi-byte determinan la longitud de la secuencia.
- Los bytes restantes en una secuencia multi-byte tienen 10 como sus 2 bits más significativos.

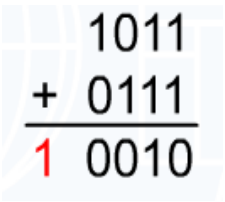
Rango UNICODE (hexadecimal)	UTF-8 secuencia de octetos (binario)
0000 0000-0000 007F	0xxxxxxx
0000 0080-0000 07FF	110xxxxx 10xxxxxx
0000 0800-0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000-0010 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
0020 0000-03FF FFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
0400 0000-7FFF FFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

Resumiendo, el código Unicode hay que entenderlo como una gigantesca base de datos con todos los símbolos utilizados por todas las lenguas del mundo, siendo algunos de ellos combinables entre si.

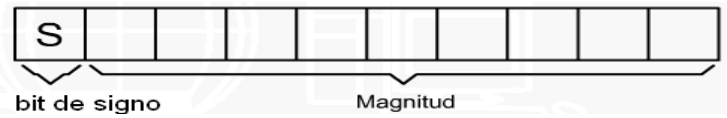
2.4. Sistemas de representación numérica

La información numérica está formada por números que representan cantidades o valores algebraicos con los que se pueden hacer operaciones matemáticas. Algunas técnicas de codificación son las siguientes:

Binario puro o coma fija: representa los números enteros positivos, con n bits se tiene un rango de representación de [0 , 2^n – 1], por ejemplo con 4 bits existe un rango de representación desde el 0 hasta el 15. Toda operación que de un resultado superior a 2^n – 1, con n bits, no se podrá realizar correctamente.



Signo magnitud: permite representar números positivos y negativos. El bit más significativo indica el signo, un 1 representa que el número es negativo. Tiene los inconvenientes de que el cero tiene dos representaciones y que para hacer cualquier operación es necesario realizar un análisis previo del signo.



Representación en exceso: suma al número a representar una constante para que el número sea siempre positivo, generalmente se suma 2^{n-1}. Por ejemplo, con 8 bits en binario puro se puede representar los números del 0 al 255, en la representación en exceso la constante sería $2^7 = 128$, con lo que se podría representar los números del -128 al 127. Algunos ejemplos:

-45 = 128 + (-45) = 83 = 01010011
+45 = 128 + 45 = 173 = 10101101
127 = 128 + 127 = 255 = 11111111

Representación de complementos: este sistema de representación sirve para tratar las restas como operaciones de suma. Un solo circuito servirá para sumar y restar. El complemento a la base b de un número N compuesto de n bits se define como: $b^n - N$ (¡¡hace falta restar!!).

Para realizar el **complemento a 1** en binario los 1 pasan a 0 y los 0 a 1. *Ejemplo: Ca1 de 101101 = 010010*

El **complemento a 2** es igual al complemento a 1 más 1. *Ejemplo: Ca2 de 0110 → Ca1 = 1001 + 1 = 1010.* Si se vuelve a hacer el Ca2 se obtiene el número original: 1010 → Ca1 = 0101 + 1 = 0110. *"TRUCO:" para hacer el complemento a 2 más rápido: copiar los bits desde la derecha hasta el primer 1 (incluido) y continuar invirtiendo los 1 por 0 y los 0 por 1.*

Y, por fin, vamos a ver cómo facilita la resta el complemento. La resta binaria de dos números puede obtenerse sumando al minuendo el complemento a dos del sustraendo. Veamos un ejemplo: la resta, 91 – 46 = 45, en binario: *1011011 – 0101110 = 0101101*
Esta misma resta puede hacerse como una suma, utilizando el complemento a dos del sustraendo: *1011011 + 1010010 = 0101101*

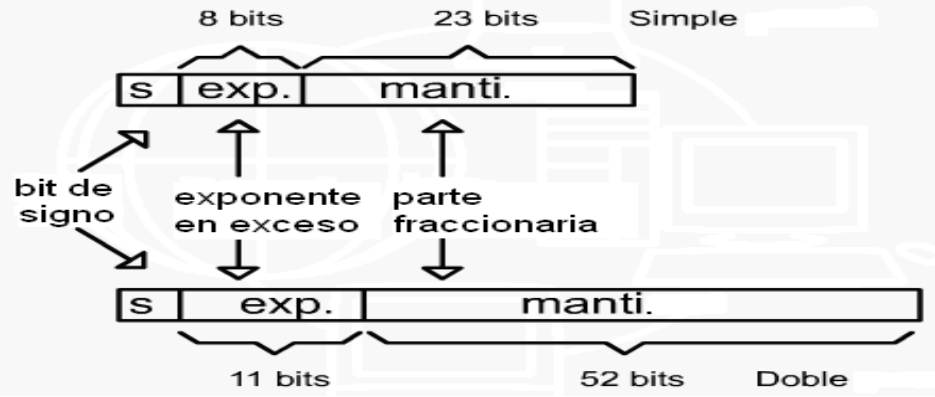
En el resultado de la suma nos sobra un bit, que se desborda por la izquierda. Pero, como el número resultante no puede ser más largo que el minuendo, el bit sobrante se desprecia.

Representación BCD: se codifica cada dígito decimal por separado con cuatro bits. Ejemplo: 43 = 0100 0011
Existen dos modalidades de codificación BCD la empaquetada y la desempaquetada.

Notación científica o coma flotante: se utiliza para representar números reales y enteros con un rango y precisión mayor de la que ofrece la coma fija. Para representar los números se utiliza la notación científica, donde cualquier número N se puede representar de la forma $\mathbf{N = M \times B^E}$. Donde M es la mantisa, B es la base y E el exponente, por ejemplo: N = 3498,3425 = 3498,3425 x 10^0 = 34,983425 x 10^2 = 0,34983425 x 10^4 = 34983425 x 10^{-4}

La normalización IEEE-754 para la representación de números reales: se puede presentar en dos formatos básicos:

- El de simple precisión, que utiliza 32 bits.
- El de doble precisión, que utiliza 64 bits.



Pasos para realizar la normalización IEEE-754:

- Como el sistema es binario la **base siempre es 2**.
- El campo **s** toma el valor 1 para negativos y el valor 0 para positivos.
- El **exponente** se representa en exceso $2^{n-1} – 1$. Para el formato de simple precisión el exceso es 127 y para el de doble precisión 1023.
- A la **mantisa** siempre se le supone que tiene un 1 a la izquierda, que no hace falta representar. La mantisa siempre será 1'.....

Ejemplo: 245'59375 representado en IEEE-754 de simple precisión

1) Se pasa todo a binario → 11110101'10011
2) Normalizamos → (1')111010110011 x 2^7
3) Signo → 0 (positivo)
4) Exponente en exceso → 127 + 7 = 134 → 10000110
5) Resultado → 0 10000110 11101011001100000000000

Como casos especiales si el exponente es 0 y la mantisa es 0 se está representando el número 0.

2.5. Unidades de medida

El **bit** es la unidad mínima de almacenamiento empleada en informática o en cualquier dispositivo digital. Con él podemos representar 2 valores cualesquiera; verdadero o falso, activo o inactivo, cero o uno, encendido (1) o apagado (0),...

Hoy en día utilizamos múltiplos de dicha unidad, empezando por el **byte**, que esta compuesto por **8 bits**. Tradicionalmente, en informática se han estado utilizando potencias de 2 para representar las medidas de la información; sin embargo se está extendiendo el uso de potencias de 10 (**Sistema Internacional de Medidas**).

Nombre	Simbolo	Potencias binarias y valores decimales
byte	b	$2^0 = 1$
Kbyte	KB	$2^{10} = 1\,024$
Megabyte	MB	$2^{20} = 1\,048\,576$
Gigabyte	GB	$2^{30} = 1\,073\,741\,824$
Terabyte	TB	$2^{40} = 1\,099\,511\,627\,776$
Petabyte	PB	$2^{50} = 1\,125\,899\,906\,842\,624$
Exabyte	EB	$2^{60} = 1\,152\,921\,504\,606\,846\,976$
Zettabyte	ZB	$2^{70} = 1\,180\,591\,620\,717\,411\,303\,424$
Yottabyte	YB	$2^{80} = 1\,208\,925\,819\,614\,629\,174\,706\,176$

Decimal	Binary	Binary is larger by...
Kilobyte KB = 10^{3} bytes	Kibibyte KiB = 2^{10} bytes	2%
Megabyte MB = 10^{6} bytes	Mebibyte MiB = 2^{20} bytes	5%
Gigabyte GB = 10^{9} bytes	Gibibyte GiB = 2^{30} bytes	7%
Terabyte TB = 10^{12} bytes	Tebibyte TiB = 2^{40} bytes	10%
Petabyte PB = 10^{15} bytes	Pebibyte PiB = 2^{50} bytes	13%

3. LÓGICA DE CIRCUITOS

El funcionamiento de los ordenadores se basa en la memorización y procesamiento de datos binarios. Para ello los ordenadores disponen de elementos básicos de **memoria** que pueden almacenar dos posibles estados y de **circuitos** que pueden operar con los datos binarios bajo la acción de **señales de control**.

Los **circuitos combinacionales y secuenciales** se componen de **puertas lógicas** cuyo funcionamiento está basado en el **Álgebra de Boole** que es el fundamento matemático de la lógica digital.

3.1. Álgebra de Boole

La circuitería de los ordenadores digitales se diseña y analiza con el uso de una disciplina matemática denominada álgebra de Boole. El álgebra de Boole es una herramienta útil tanto en el análisis como en el diseño de circuitos digitales.

En el álgebra de Boole existen solamente dos valores o resultados posibles, el valor lógico “cierto” y el valor lógico “falso”, contrarios entre sí y que pueden ser denotados como 1 y 0. En los circuitos lógicos electrónicos la representación de estos dos estados suele estar asociado a la presencia o ausencia de tensión. Se denomina **variable lógica** a toda variable de este álgebra que sólo puede adoptar uno de los dos valores booleanos.

Una **función boolena lógica** es una función que se define como una combinación adecuada de variables lógicas relacionadas entre sí a través de **operadores lógicos** siguiendo unas determinadas reglas de construcción. Una función lógica puede escribirse en forma de **ecuación lógica**.

La **tabla de verdad** de una función lógica es una representación del comportamiento de la misma, dependiendo de los valores particulares que pueden tomar cada una de sus variables. En ella deben figurar **todas las combinaciones posibles** entre las variables, y para cada una aparecerá el valor de la función. Así para **n variables** tendremos **2^n combinaciones** posibles.

A continuación vamos a ver las funciones lógicas básicas, el orden de ejecución de los operandos es NOT, AND y OR, para alterar este orden se utilizan los paréntesis.

La **función OR** es representada por "+" y por "v". Su valor es 1 si lo es uno cualquiera de sus operandos o los dos.

La **función AND** es representada por "·" y por "^". Su valor es 1 si ambos operandos lo son.
La **función NOT** representada por "¬", por " ' " o por " ¯ " (un guión encima de la variable), opera sobre una sola variable, siendo su valor el contrario de la variable.

A	B	$\bar{A}$	A · B	A + B
		NOT A	A AND B	A OR B
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

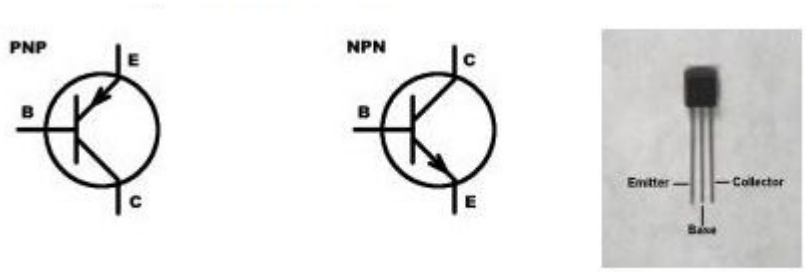
En el análisis de funciones lógicas, sobre todo para su simplificación, es necesario el conocimiento de las siguientes **propiedades del álgebra de Boole**.

Postulados básicos del álgebra de Boole		
$A * B = B * A$	$A + B = B + A$	Ley conmutativa
$A* (B+C) = (A*B) + (A*C)$	$A + (B*C) = (A+B) * (A+C)$	Ley distributiva
$A * (B * C) = (A * B) * C$	$A + (B + C) = (A + B) + C$	Ley asociativa
$0 * A = 0$	$1 + A = 1$	Elemento neutro
$1 * A = A$	$0 + A = A$	Elemento neutro
$A * \bar{A} = 0$	$A + \bar{A} = 1$	Complementación
$\overline{\overline{A}} = A$	$\overline{\overline{B}} = B$	Involución
$A * A = A$	$A + A = A$	Idempotencia
Teoremas		
$A * (A + B) = A$	$A + (A * B) = A$	Absorción
$\overline{A * B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} * \overline{B}$	Leyes de Morgan

3.2. El transistor

La base de toda la electrónica digital se encuentra en el funcionamiento de un componente electrónico llamado **transistor.**

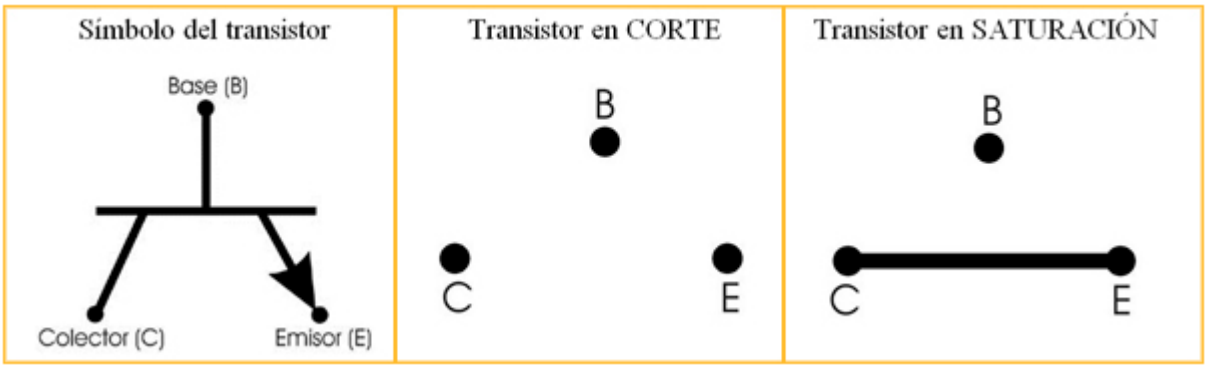
El transistor es un dispositivo semiconductor que permite el control y la regulación de una corriente grande mediante una señal muy pequeña. Existe una gran variedad de transistores. En principio, se explicarán los bipolares. Los símbolos que corresponden a este tipo de transistor son los siguientes:



El transistor es un dispositivo electrónico activo de tres terminales que puede trabajar en tres modos distintos (Zona Activa, Corte y Saturación). En la electrónica analógica se utiliza el transistor en la zona activa, utilizándose normalmente por sus propiedades de amplificación de señal.

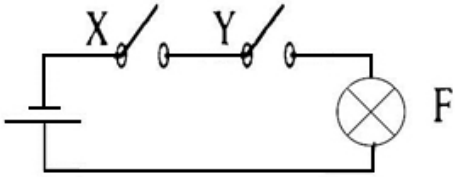
Sin embargo cuando se trabaja en electrónica digital se utiliza en corte o saturación. El funcionamiento en este caso consiste en el hecho de que según como actuemos en uno de los terminales el transistor se coloca en corte o en saturación.

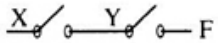
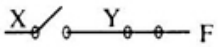
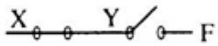
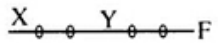
Si lo tenemos en corte, entre los otros dos terminales se establecerá un circuito abierto, mientras que si lo tenemos en saturación el circuito pasará a estar cerrado. Gráficamente:



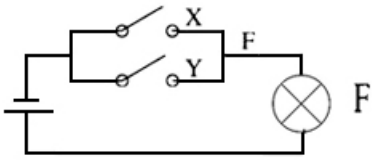
Es decir, tenemos un interruptor entre los puntos C y E controlado eléctricamente, desde B, y que puede pasar de abierto a cerrado con una señal eléctrica de manera muy rápida.

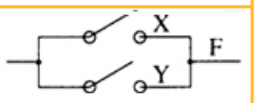
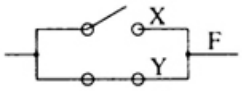
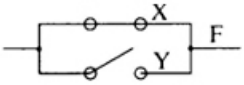
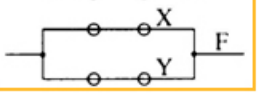
Paras simplificar la representación de los sistemas vamos a hacer una analogía asociando el símbolo del interruptor abierto al estado en corte o "0" del transistor y el interruptor cerrado al estado de saturación o "1". Podemos hacer dos sencillos montajes eléctricos:



X	Y	F	representación interruptores
0	0	0	
0	1	0	
1	0	0	
1	1	1	

En la imagen anterior vemos cómo se produce señal (luz en la bombilla “F”) si actuamos sobre el interruptor X “y” el Y.



X	Y	F	representación interruptores
0	0	0	
0	1	1	
1	0	1	
1	1	1	

En este caso se produce señal indistintamente cuando actuamos sobre X “o” sobre el Y.

Si ahora pensamos que esos interruptores pueden ser transistores trabajando en corte o saturación tenemos lo que se conoce con el nombre de **puertas lógicas.** De modo análogo, utilizando el Álgebra de Boole, que es la base matemática sobre en la que se sustentan las técnicas digitales, podemos hacer puertas o componentes electrónicos que sumen, multipliquen, comparen cantidades, etc., o que lo hagan todo, como es el caso de la ALU, elemento que veremos forma parte del procesador de un ordenador.

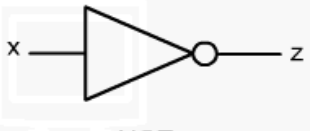
3.3. Puertas lógicas

Las puertas lógicas son **dispositivos electrónicos** diseñados para realizar operaciones elementales con la información binaria, es decir, con tensión en sus entradas o con ausencia de ésta.

Haciendo agrupaciones de puertas lógicas se pueden construir circuitos elementales, que unidos entre si permitirá obtener circuitos cada vez más complejos, las propiedades del álgebra de Boole se utilizan sobre todo para la simplificación de circuitos aunque hay otros métodos que automatizan el proceso como el método de Quine-McCluskey o los mapas de Karnaugh.

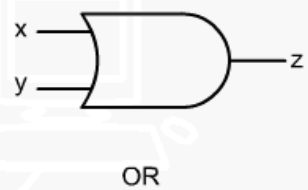
Puerta NOT (inversora): ejecuta la función lógica básica llamada inversión o complementación. Se trata de una operación que solo maneja una variable de entrada y otra de salida. La salida toma el estado opuesto al que tiene la entrada. En definitiva, en lo que a bits respecta, cambia de 1 a 0 y de 0 a 1. La ecuación que la representa es $X = \overline{Z}$.

Entrada x	Salida z
1	0
0	1



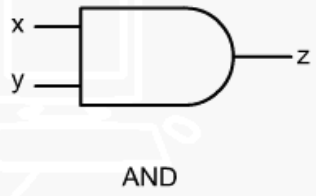
Puerta OR (suma lógica): cuando distintas variables lógicas se combinan mediante la función OR, el resultado toma el valor 1 si alguna de ellas tiene dicho estado. La ecuación que representa la función OR de dos variables de entrada es: Z = X + Y.

Entrada x	Entrada y	Salida z
0	0	0
0	1	1
1	0	1
1	1	1



Puerta AND (producto lógico): cuando varias variables lógicas, de tipo binario, se combinan mediante la operación lógica AND, producen una variable de salida, que solo toma el nivel lógico 1, si todas ellas tienen dicho nivel o estado. La ecuación lógica de la función AND para dos variables de entrada es: $Z = X \cdot Y$.

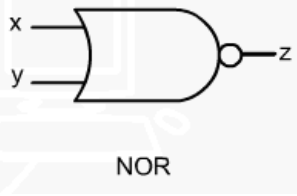
Entrada x	Entrada y	Salida z
0	0	0
0	1	0
1	0	0
1	1	1



AND

Puerta NOR (suma lógica inversa): Esta puerta produce la función inversa de la puerta OR, es decir, la negación de la suma lógica de las variables de entrada. Su comportamiento es equivalente a la de la puerta OR seguida de una NOT. La ecuación lógica es la siguiente: $Z = \overline{X + Y}$

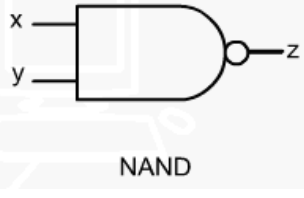
Entrada x	Entrada y	Salida z
0	0	1
0	1	0
1	0	0
1	1	0



NOR

Puerta NAND (producto lógico inverso): La puerta NAND produce la función inversa de la AND, o sea, la negación del producto lógico de las variables de entrada. Actúa como una puerta AND seguida de una NOT. La ecuación lógica es la siguiente: $Z = \overline{X \cdot Y}$

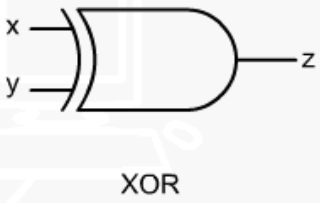
Entrada x	Entrada y	Salida z
0	0	1
0	1	1
1	0	1
1	1	0



NAND

Puerta XOR (OR exclusiva): La salida de esta puerta es 1 si solo una de las entradas es 1. La ecuación lógica es la siguiente: $Z = X \oplus Y$

Entrada x	Entrada y	Salida z
0	0	0
0	1	1
1	0	1
1	1	0



XOR

Tabla resumen de las puertas lógicas:

entradas A	B	AND	NAND	OR	NOR	XOR
0	0	0	1	0	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	0	0
funciones		$A \cdot B$	$\overline{A \cdot B}$	$A + B$	$\overline{A + B}$	$A \oplus B$

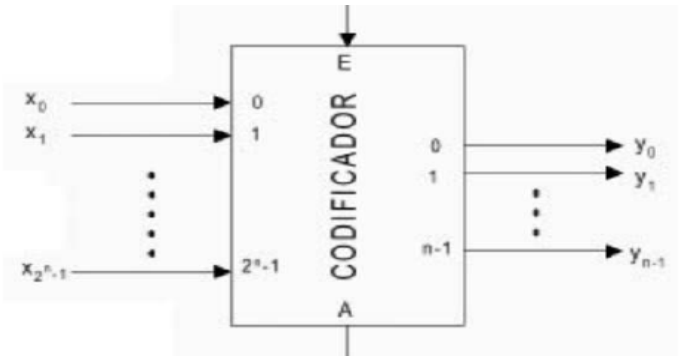
Físicamente una puerta lógica se diseña con un circuito electrónico. Las puertas lógicas no se fabrican ni se venden individualmente, sino en unidades llamadas **circuitos integrados** o **chips**. Las puertas lógicas se utilizan como elemento base para la construcción de circuitos en electrónica digital.

3.4. Circuitos combinacionales

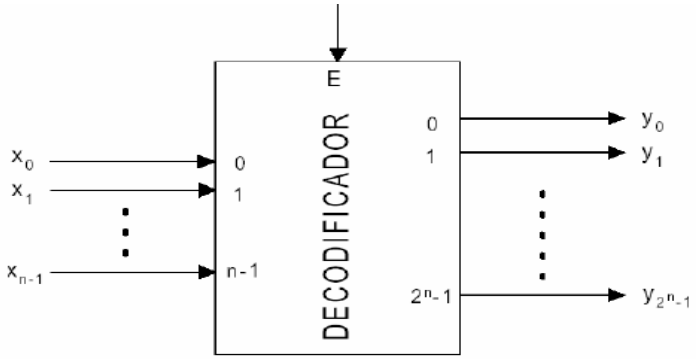
Un circuito combinacional es un conjunto de puertas interconectadas, cuya **salida depende únicamente del valor de las entradas en ese instante**, teniendo en cuenta que puede existir un retardo entre las entradas y las salidas. Es decir, es un circuito sin elementos de memoria, los valores anteriores no influyen en el valor actual de la salida.

Un circuito combinacional consta de tres elementos básicos: variables de entrada, variables de salida y circuito interno formado por puertas lógicas. A continuación se estudian algunos circuitos que responden a funciones lógicas muy utilizadas y que suelen presentarse ya integradas en circuitos comerciales.

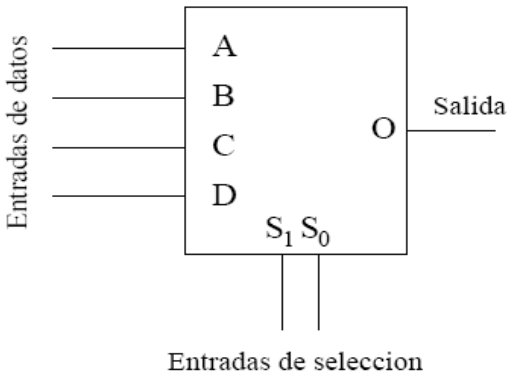
- **Codificador:** Un codificador es un circuito combinacional con un conjunto de entradas 2^N y un número de salidas N, cuyo propósito es mostrar en la salida el código binario correspondiente a la entrada activada.



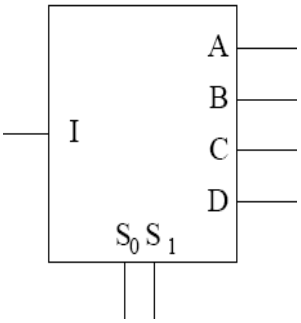
- **Decodificador:** Los decodificadores efectúan la operación inversa de los codificadores. Disponen de un conjunto N de entradas y un conjunto 2^N de salidas. Cuando aparece un código binario a la entrada, se activa la salida identificada con el número decimal equivalente.



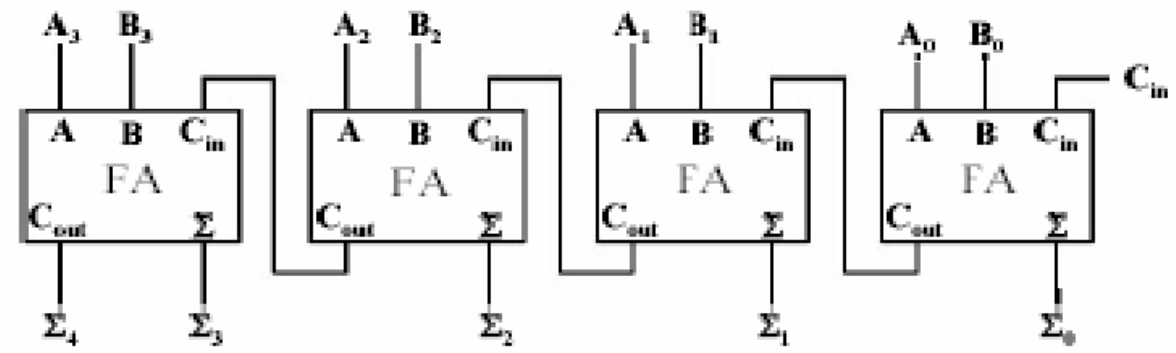
- **Multiplexor:** son circuitos con 2^n líneas de entrada, n líneas de selección y una línea de salida. Su objetivo es colocar en la salida el dato presente en la línea de entrada seleccionada por las líneas de control.



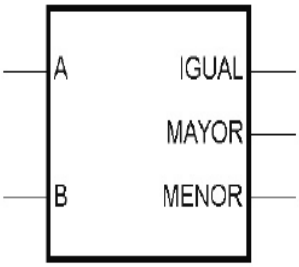
- **Demultiplexor:** realiza la función contraria a la de los multiplexores. Tiene una entrada cuyo valor se representa en una de sus salidas, la cual será seleccionada por las líneas de control.



- **Sumador:** Para construir sumadores completos (Full Adder) es necesario considerar en cada suma parcial de dos bits el posible acarreo anterior procedente de la suma parcial de los bits anteriores. A partir de un sumador completo puede construirse un sumador de cualquier número de bits concatenando sumadores completos.



- **Comparador:** con dos entradas de n bits y tres salidas, tiene por misión comparar las entradas determinando si una es mayor, igual o menor que la otra.



3.5. Circuitos secuenciales

Un sistema secuencial es aquel en cuyo comportamiento influye, no sólo la entrada en un instante dado, sino todos los valores pasados de la misma, es decir, **la salida depende de la secuencia de entradas aplicadas**. En la arquitectura de computadores, componentes básicos como los registros, memorias y la unidad de control, se construyen mediante circuitos secunciales, aunque de muy diverso grado de complejidad.

Atendiendo a la forma de comportarse en el tiempo los sistemas secuenciales pueden ser de dos tipos: **síncronos** y **asíncronos**.

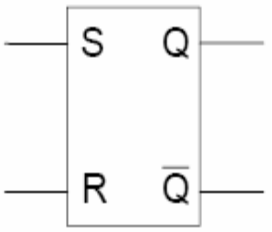
Los **síncronos** son aquellos en los que la transición de estado se produce en momentos específicos, determinados por un pulso en una señal de actuación adicional llamada de sincronización o señal de reloj. Una vez alcanzado un estado, éste permanece estable hasta el siguiente pulso de sincronización, aunque la señal de entrada varíe entre el intervalo de los pulsos.

Los **asíncronos** no tienen señal de sincronismo, las transiciones pueden producirse en cualquier instante, para lo que únicamente es necesario que cambien las entradas.

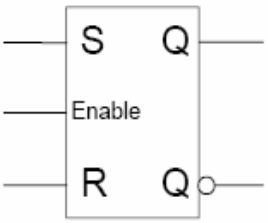
Los **biestables**, también llamados flip-flop, son circuitos capaces de retener información, son unidades de memoria que mantienen su último estado indefinidamente mientras que no se produzca un cambio en sus entradas. Existen diferentes tipos de biestables:

- **Biestable S-R asíncrono:** tiene dos entradas S (set) y R (reset), y dos salidas complementarias Q y $\overline{Q}$. El comportamiento del biestable es el siguiente:
    - Cuando ambas entradas valen 0 el estado se mantiene.
    - Cuando S=1, entonces Q=1 y se mantendrá aunque S pase a 0.
    - Cuando R=1, entonces Q=0 y se mantendrá aunque R pase a 0.
    - Cuando R=1 y S=1, Q tendrá un valor no definido.

  En resumen, en el biestable S-R, la activación de la señal S hace que se coloque un 1 en la salida, mientras que la activación de la señal R hace que se coloque un 0.



- **Biestable S-R síncrono:** activado por nivel o por flanco (cambio de estado) tiene la característica de que para que los valores presentados en sus entradas tengan efecto en sus salidas es necesario que la señal de reloj se encuentre activa.



- **Biestable J-K asíncrono:** es similar al S-R, pero eliminando la indeterminación que éste presenta, así cuando sus dos entradas se encuentran a 1 cambia al estado opuesto.

- **Biestable J-K síncrono:** similar al biestable J-K asíncrono pero activado por una señal de reloj.

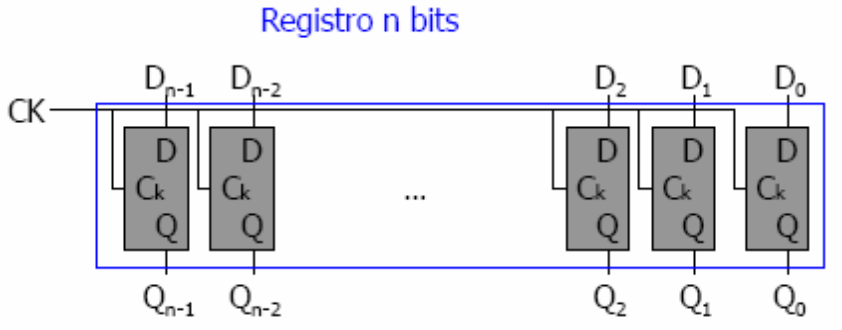
- **Biestable D:** es siempre síncrono, posee una entrada de información D, una de sincronismo CK y dos salidas complementarias Q y $\overline{Q}$. La salida Q toma el valor de la entrada D mientras la señal de sincronismo CK está a nivel alto, y cuando ésta pasa a nivel bajo, la salida Q se mantiene independientemente del valor de la entrada D.

- **Biestable T asíncrono:** tiene una única entrada T y dos salidas complementarias Q y $\overline{Q}$. Cada vez que T vale 1 se invierte el valor de Q, cuando T vale 0 la salida Q mantiene su valor.

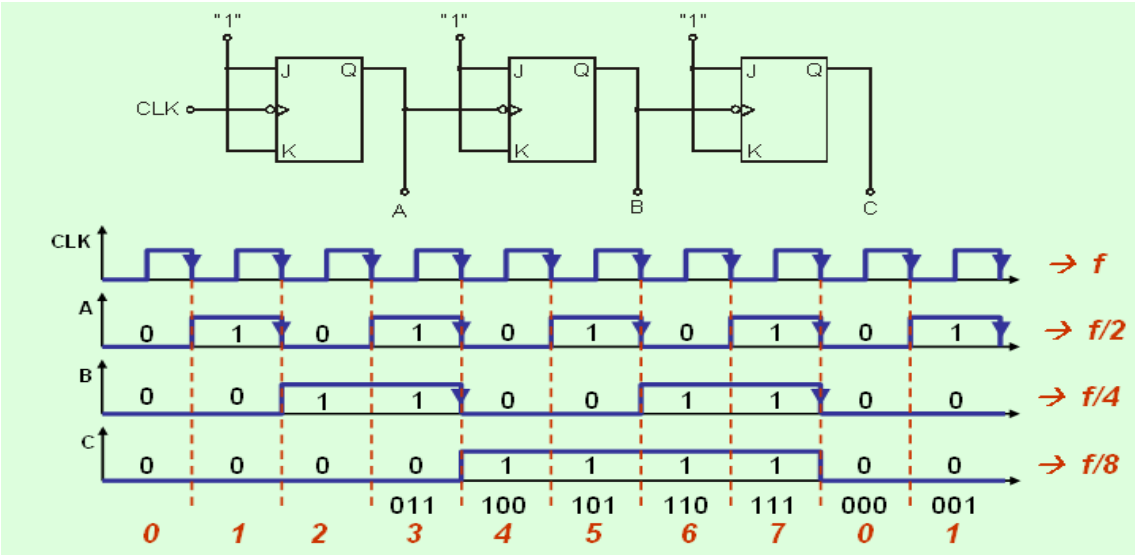
- **Biestable T síncrono:** la función de este dispositivo es la de cambiar su estado cada vez que así lo indique la señal de reloj.

Algunos circuitos secuenciales que pueden utilizarse como bloques para el diseño de otros circuitos secuenciales son los siguientes:

- **Registros:** es un circuito secuencial síncrono capaz de almacenar varios bits de información. Con n biestables D se pueden construir registros de n bits.



- **Contadores:** Un contador es un circuito hecho con biestables JK. Tienen una entrada de reloj genérica y n salidas binarias que representan en cada momento el valor de la cuenta en binario de los pulsos que entran por la entrada de reloj.



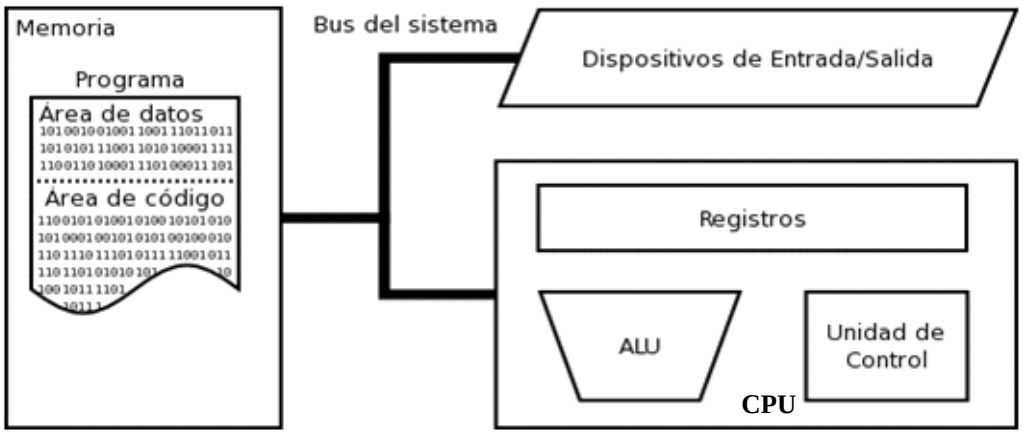
4. COMPONENTES FÍSICOS DE LOS SISTEMAS INFORMÁTICOS

4.1. Estructura básica de un ordenador

El modelo básico de arquitectura empleada en los ordenadores digitales fue establecido en 1946 por John Von Neumann. La idea de Von Neumann consistió en conectar permanentemente las unidades del ordenador, siendo coordinado su funcionamiento bajo un control central.

Desde un punto de vista físico, esta máquina esta compuesta por cuatro componentes básicos o unidades funcionales:

- **Unidad central de proceso** (CPU): que se compone de unidad de control (UC), unidad aritmético-lógica (ALU), y registros.
- **Memoria principal:** empleada para almacenar tanto datos como instrucciones máquina.
- **Unidad de entrada/salida:** realiza la transferencia de información con los periféricos.
- **Buses:** que son caminos a través de los cuales las instrucciones y los datos circulan entre las distintas unidades del ordenador.



4.2. La unidad central de proceso (CPU)

La unidad central de proceso (CPU), es el conjunto formado por la unidad de control (UC), los registros y la unidad aritmético-lógica (ALU) de un ordenador. La CPU trabaja interpretando y ejecutando las instrucciones contenidas en los programa.

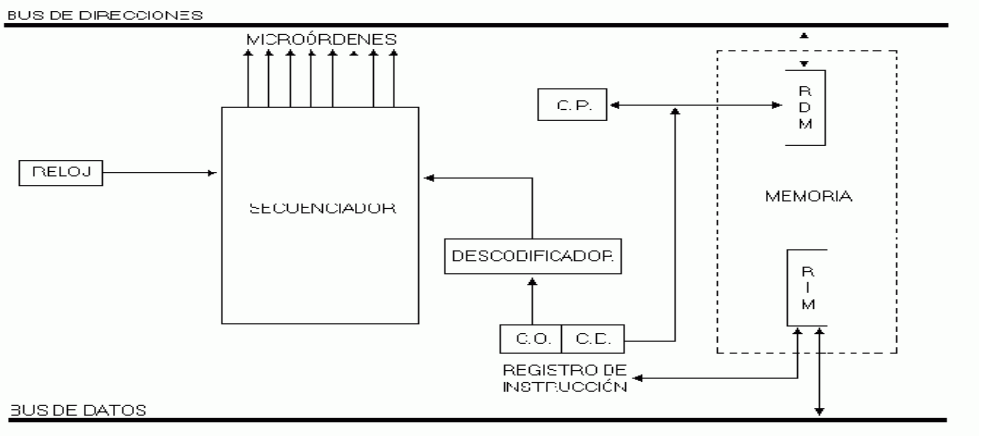
Los ordenadores no son capaces de interpretar directamente un lenguaje de programación de alto nivel, únicamente son capaces de interpretar un lenguaje muy restringido llamado **lenguaje máquina**. Este lenguaje es muy sencillo y se compone de una serie de instrucciones máquina cuyo conjunto constituye el llamado **juego de instrucciones** del ordenador.

La unidad de control (UC)

El objetivo de la UC es monitorizar el funcionamiento de todo el ordenador. La unidad de control dirige y coordina todas las operaciones que tienen lugar en las restantes unidades, además de interpretar y ejecutar las instrucciones controlando su secuencia.

La UC realiza sus funciones generando **señales de control** que producen determinadas operaciones en un orden de forma sincronizada por un temporizador o **reloj**.

Se pueden utilizar dos metodologías para diseñar la UC, la **lógica cableada** basada en la utilización de puertas lógicas, o la **microprogramación** que se basa en almacenar en una memoria micro órdenes: la ejecución de una instrucción implica leer de la memoria central las micro órdenes correspondientes.

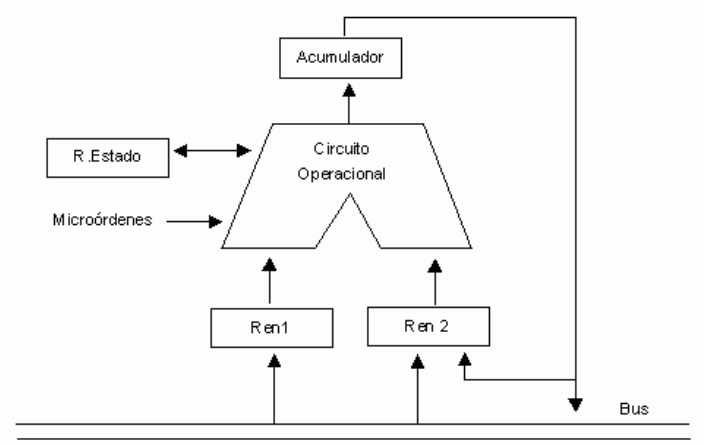


Para realizar sus funciones, la UC consta de los siguientes elementos:

- **Reloj:** proporciona una sucesión de impulsos eléctricos a intervalos constantes (frecuencia constante) que marcan los instantes en que han de comenzar los distintos pasos de que consta cada instrucción.
- **El registro de instrucción (R.I.):** almacena la instrucción del programa que se está ejecutando en cada momento. Este registro suele estar dividido en campos. Cada campo contiene un número de bits variable con cada arquitectura. Una instrucción de un programa se compone normalmente de dos partes: código de operación y código de datos.
- **Decodificador:** extrae el código de operación de la instrucción en curso del RI, lo analiza y emite las señales necesarias al resto de elementos para su ejecución a través del secuenciador.
- **Contador de programa (CP):** Contiene permanentemente la dirección de memoria de la siguiente instrucción a ejecutar.
- **El secuenciador:** Es el verdadero centro de operaciones del ordenador. Es el dispositivo en el que se generan órdenes muy elementales (microórdenes) que sincronizadas por los impulsos del reloj hacen que se vaya ejecutando, poco a poco, la instrucción que está cargada en R.I.

La unidad aritmético-lógica (ALU)

La ALU es la encargada de tratar los datos, ejecutando las operaciones requeridas por la unidad de control. La ALU se encarga de realizar las operaciones elementales de tipo aritmético (sumas, restas, productos, divisiones) y de tipo lógico (comparaciones), requeridas para la ejecución de los programas.



Para realizar su función, la ALU necesita de los siguientes elementos:

- **Circuito operacional o combinacional (COP):** Se encarga de realizar las operaciones con los datos procedentes de los registros de entrada (REN1, REN2). Este circuito tiene unas entradas de órdenes para seleccionar o indicarle el tipo de operación que se desea realizar con los datos de entrada (suma, resta, etc).

- **Registros de entrada (REN):** REN1 y REN2 se encargan de almacenar los dos operandos de entrada que intervienen en una instrucción antes de la realización de la operación por parte del circuito operacional. También se emplean para el almacenamiento de resultados intermedios o finales de las operaciones respectivas.

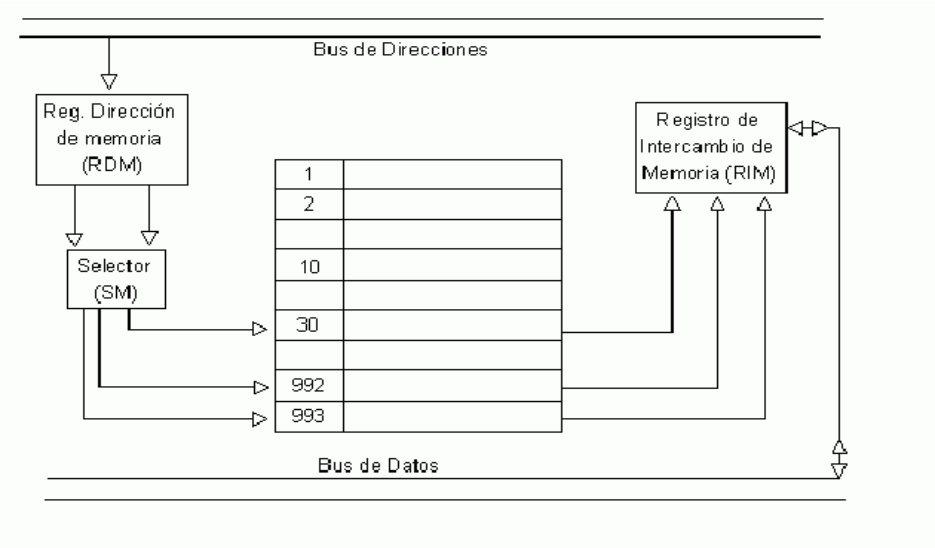
- **Registro acumulador o registro de resultado (RA):** Sirve como registro de almacenamiento de los resultados de las operaciones llevadas a cabo por el circuito combinacional. Está conectado al registro de entrada REN2 para realimentación en el caso de operaciones encadenadas, se guardan los resultados intermedios en el acumulador. Asimismo, tiene una conexión directa al bus de datos para el envío de los resultados a la memoria central o a la unidad de control.

- **Registro de Estado (RES):** este registro está formado por un conjunto de biestables (son circuitos capaces de retener información, es decir, son unidades de memoria que mantienen su último estado indefinidamente mientras no se produzca un cambio en sus estados). Contiene información sobre el resultado de la última operación realizada y se tiene en cuenta en operaciones posteriores. Cada uno de estos biestables señala una determinada condición sobre el último valor que ha sido escrito en el acumulador, los más típicos son los siguientes:

    - Z (bit de cero). Si en la última operación de la ALU el resultado ha sido cero o no.
    - N (bit de negativo). Adquiere el valor uno cuando el último número binario escrito en el acumulador es negativo.
    - C (bit de acarreo). Se pone a uno cuando después de efectuar una suma o una resta, ha habido acarreo.
    - O (bit de desbordamiento). Se activa si la operación genera un resultado fuera de límites.

4.3. La memoria principal

La memoria principal no es nada más que un conjunto ordenado de **celdas** o posiciones de memoria, numeradas de forma consecutiva, capaces de retener información, mientras el ordenador está conectado. A cada celda se puede acceder por medio de un número que la identifica. Dicho número se conoce con el nombre de **dirección de memoria**. Mediante esta dirección se puede acceder de forma directa a cualquier posición, se dice, por ello, que la memoria central es un soporte de información de acceso directo.



No hay que confundir los términos celda o posición de memoria con el de **palabra**, ya que esta última es la cantidad de información que puede introducirse o extraerse de la memoria central de una sola vez. Información que se puede leer o escribir en un único golpe de reloj. El tamaño habitual de la palabra suele ser 16, 32 o 64 bits.

La memoria central tiene asociados dos registros para la realización de operaciones de lectura o escritura y un dispositivo encargado de seleccionar una celda de memoria en cada operación de acceso a la misma:

- **Registro de Dirección de Memoria (RDM):** antes de la realización de una operación de lectura o escritura se ha de colocar en este registro la dirección de la celda que se va a utilizar en la operación, bien para grabar en ella o para extraer de la misma el dato correspondiente.

- **Registro de Intercambio de Memoria (RIM):** si se trata de una operación de lectura de memoria, este registro (RIM) es el que recibe el dato de la memoria señalado por el registro RDM para su envío por medio del bus de datos a uno de los registros de la ALU o a la unidad que lo requiera.

  Si se trata de una operación de escritura en memoria, la información que hay que grabar, es depositada en el RIM para que desde él se transfiera a la posición de memoria indicada por el RDM.

- **Selector de Memoria (SM):** este dispositivo se activa cada vez que se produce una orden de lectura o escritura, conectando la celda de memoria, cuya dirección figura en el RDM, con el RIM y posibilitando la transferencia de los datos en un sentido u otro.

4.4. Buses de comunicación

La CPU se comunica o conecta con las unidades que integran el sistema por medio de buses o grupos de líneas. Los buses se pueden dividir en tres tipos dependiendo de las instrucciones y datos que transportan: bus de datos, bus de direcciones y bus de control.

- **Bus de Datos:** permite la circulación de valores entre registros. Es utilizado por la CPU para realizar el intercambio de instrucciones y datos con el exterior. A través de estas conexiones se efectuará la transferencia de información.
- **Bus de Direcciones:** consiste en un canal constituido por líneas de direcciones que indican la posición de memoria en la que se encuentra la información o del periférico a tratar. Una vez direccionada la posición, la información almacenada pasará a la CPU a través del bus de datos. La anchura del bus de direcciones indica la cantidad de memoria a la que puede acceder un procesador.
- **Bus de Control:** está formado por un número variable de líneas eléctricas a través de las cuales controla a las unidades complementarias. El procesador proporciona unas señales para sincronizar las selecciones de posiciones de memoria y la transferencia de datos.

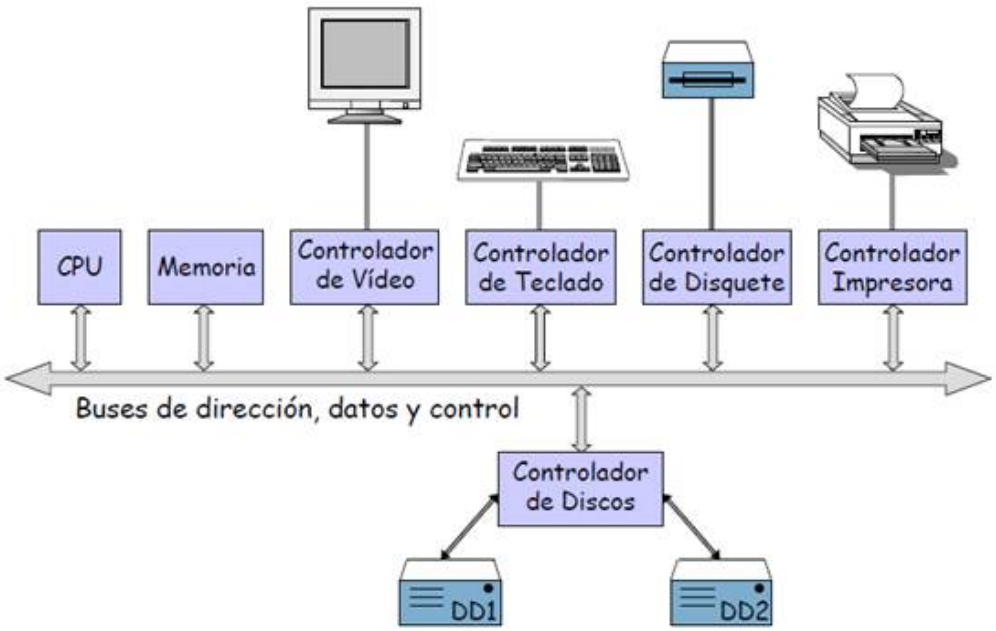
4.5. Unidades de entrada/salida

El concepto de entrada y salida hace referencia a toda comunicación o intercambio de información entre la CPU o la memoria principal con el exterior. La parte del ordenador que permite esta comunicación es la unidad de entrada/salida.

En el sistema de entrada/salida se encuentran dos partes fundamentales, los periféricos y la interfaz. Los **periféricos** son dispositivos electromecánicos que permiten la comunicación directa con el mundo exterior, existen periféricos de memoria secundaria y auxiliar, periféricos de entrada y/o salida de datos, y periféricos de comunicación de datos. La **interfaz** o **controladora** es el conjunto de circuitos y programas que se utilizan para permitir la comunicación entre el periférico y la CPU o memoria, salvando diferencias como velocidades de transmisión o formato de los datos.

Para que se pueda llevar a cabo el intercambio de información se deben llevar a cabo las siguientes tareas:

- **Direccionamiento**: selección del dispositivo de entrada/salida implicado en una transferencia determinada.
- **Transferencia:** intercambio de datos desde o hacia el dispositivo seleccionado.
- **Sincronización:** entre los periféricos y la CPU.



4.6. Esquema básico de funcionamiento de un ordenador

Para que un ordenador pueda ejecutar un programa, éste ha de estar almacenado en la memoria principal. La unidad central de proceso tomará una a una sus instrucciones e irá realizando las tareas correspondientes. Al conjunto de acciones que se llevan a cabo en la realización de una instrucción se denomina **ciclo de instrucción**, que se compone de dos fases:

- **Fase de búsqueda o carga**: en esta fase se transfiere la instrucción correspondiente desde la memoria central a la unidad de control.
- **Fase de ejecución:** en esta fase se realizan todas las acciones que conlleva la propia instrucción.

Fase de carga de una instrucción.

Suponiendo que se tiene un ejemplo de instrucción aritmética de suma con tres direcciones y direccionamiento directo; es decir, la instrucción contiene el código de operación correspondiente a la suma, los dos primeros operandos están en las direcciones de memoria correspondiente y el resultado ha de quedar en la dirección indicada por el tercer operando.

EJEMPLO: SUMAR 033 992 993 (Sumar los contenidos de memoria 33 y 992 y almacenar el resultado en la posición 993).

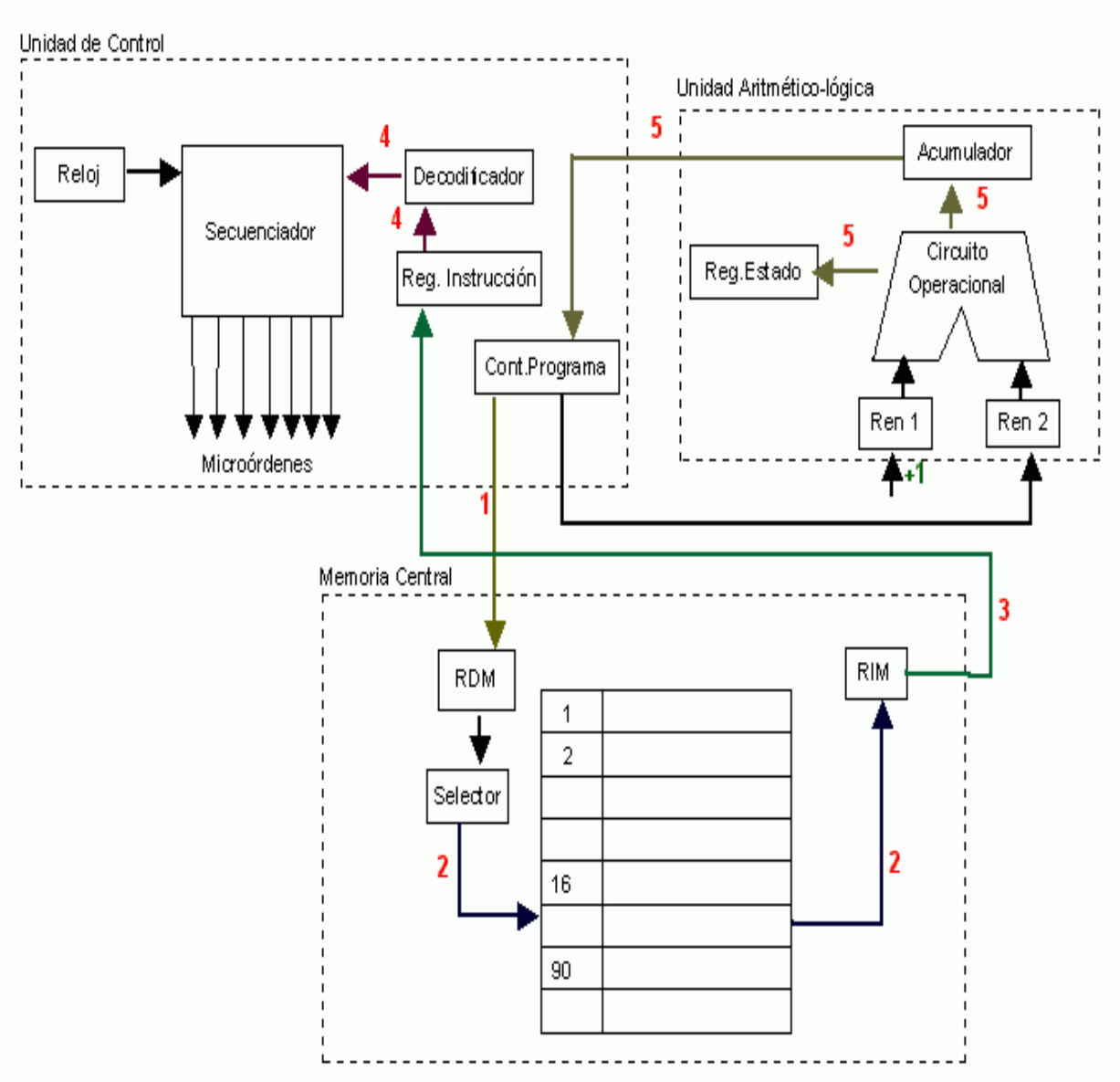
PASO 1.- La UC envía una microórden para que el contenido del registro de contador de programa (CP) que contiene la dirección de la siguiente instrucción (instrucción que corresponde procesar), sea transferido al registro de dirección de memoria (RDM).

PASO 2.- La posición de memoria que figura en el registro de dirección de memoria (RDM) es utilizada por el Selector para transferir su contenido (instrucción) al registro de intercambio de memoria (RIM).

PASO 3.- Se transfiere la instrucción desde el registro de intercambio de memoria (RIM) al registro de instrucción (RI) de la U.C.

PASO 4.- A continuación el descodificador procede a interpretar la instrucción que acaba de llegar al registro de instrucción (RI); en este caso SUMAR, quedando dispuesto para la activación del circuito sumador de la ALU e informando al Secuenciador.

PASO 5.- El registro contador de programa (CP) se autoincrementa (utilizando la ALU) con un valor 1 (o n en el caso de que sea ésta la longitud de la palabra de memoria), de tal forma que quede apuntando a la siguiente instrucción situada consecutivamente en memoria. Si la instrucción en ejecución es de ruptura de secuencia, el contador de programa (CP) se cargará con la dirección que corresponda.



Ejecución de una instrucción.

La ejecución se realiza en los siguientes pasos, teniendo en cuenta que si la instrucción no hubiese necesitado operandos, no se ejecutarían los pasos 1 a 6 ni el 8.

PASO 1.- Se transfiere la dirección del primer operando desde el registro de instrucción (RI) al registro de dirección de memoria (RDM).

PASO 2.- El selector extrae de la memoria dicho dato depositándolo en el registro de intercambio de memoria (RIM).

PASO 3.- Se lleva este operando desde el registro de intercambio de memoria (RIM) al registro de entrada 1 (REN 1) de la unidad aritmético-lógica. (ALU).

PASO 4.- Se transfiere la dirección del segundo operando desde el registro de instrucción (RI) al registro de dirección de memoria (RDM).

PASO 5.- El selector extrae de la memoria dicho dato depositándolo en el registro de intercambio de memoria (RIM).

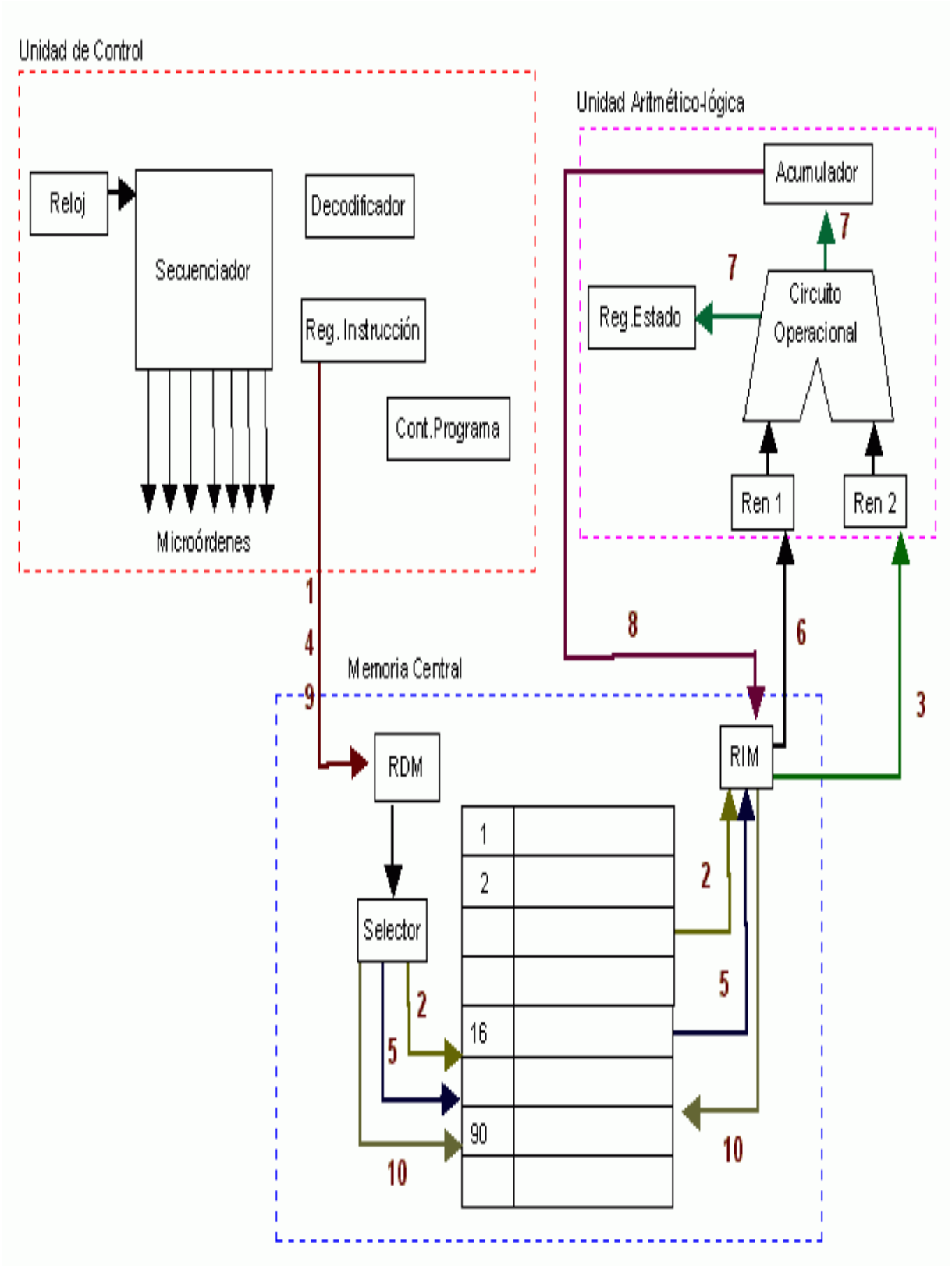
PASO 6.- Se lleva este operando desde el registro de intercambio de memoria (RIM) al registro de entrada 2 (REN 2) de la unidad aritmético-lógica (UAL).

PASO 7.- El secuenciador envía una microorden a la unidad aritmético-lógica (ALU) para que se ejecute la operación de que se trate. El resultado de la operación queda almacenado en el registro acumulador (RA).

PASO 8.- Este resultado es enviado desde el registro acumulador (RA) al registro de intercambio de memoria (RIM).

PASO 9.- Se transfiere desde el registro de instrucción (RI) al registro de dirección de memoria (RDM) la dirección donde ha de almacenarse el resultado de la memoria.

PASO 10.- Se transfiere el resultado desde el registro de intercambio de memoria (RIM) a la dirección de memoria indicada en el registro de dirección de memoria (RDM).



5. EVOLUCIÓN DE LOS SISTEMAS INFORMÁTICOS

La clasificación de los ordenadores en generaciones está basada en la forma en que los avances tecnológicos han afectado a los mismos:

- **Primera Generación** (1940 – 1952). La constituyen todos aquellos ordenadores diseñados a base de válvulas de vacío y cuyo uso principal fue dirigido a las áreas científico-militares. Utilizaban como lenguaje de programación el lenguaje máquina y como memorias para conservar la información las tarjetas o cintas perforadas.
- **Segunda Generación** (1952 - 1964). Se sustituyen las válvulas de vacío por el transistor. Ello conlleva que las máquinas ganen en potencia y fiabilidad, reduciéndose el consumo, el tamaño y el precio de las mismas. Se empiezan a utilizar los ordenadores en el campo administrativo y empiezan a utilizarse lenguajes de programación como Ensamblador, Fortran, Cobol. Como memorias se utilizan los tambores magnéticos y los núcleos de ferrita.
- **Tercera Generación** (1964 - 1971). Se caracteriza principalmente por la aparición de los circuitos integrados, que consistían en el encapsulamiento de una gran cantidad de componentes (resistencias, diodos, transistores), conformando uno o varios circuitos con una función concreta, sobre una pastilla de silicio y metal. Asimismo se produjo una gran evolución en el desarrollo de software, sobre todo a nivel de sistemas operativos, empiezan a utilizarse los discos magnéticos.
- **Cuarta Generación** (1971 - 1981). Aparece el microprocesador, que consiste en la integración de toda la CPU del ordenador en un solo circuito integrado. Evolución importante de los lenguajes de programación y sobre todo marca el inicio de las telecomunicaciones.
- **Quinta Generación** (1981 - actualidad). Alta escala de integración en circuitos integrados (VLSI). Lenguajes de programación muy próximos al lenguaje natural. Redes integradas y entornos multimedia con sonido, datos e imágenes.
- **Sexta Generación** (actualidad – futuro). Esta generación está basada en hardware de procesamiento paralelo y en software de inteligencia artificial. La inteligencia artificial es un campo emergente en la ciencia de computación, la cual interpreta los métodos necesarios para hacer que las computadores piensen como seres humanos. Se estima que la computación cuántica y la nano tecnología van a cambiar radicalmente los ordenadores del futuro.

6. SISTEMAS OPERATIVOS

Se puede definir un **sistema operativo** desde tres puntos de vista, según la visión del usuario, como máquina extendida y como gestor de recursos.

Según la **visión del usuario** un sistema operativo es el programa que actúa como interfaz entre los seres humanos y el ordenador.

Como **máquina extendida,** el sistema operativo debe ocultar la complejidad del hardware proporcionando una abstracción de mayor nivel, es decir, el sistema operativo debe hacer transparente al usuario las características hardware concretas de los dispositivos.

Como **gestor de recursos,** el sistema operativo establece la política que determina a quien, cuando, cuanto tiempo y la cantidad de recursos que asigna. El sistema operativo también debe colaborar para que el ordenador se utilice eficientemente, el caso más evidente es la capacidad de que varios procesos se alternen en el uso del procesador, ofreciendo a los usuarios la sensación de ejecución paralela.

Resumiendo, un sistema operativo es un conjunto de **programas de control** cuya finalidad es hacer más **fácil** y **eficiente** el uso del ordenador en que se ejecuta.

6.1. Arquitectura de los Sistemas Operativos

A lo largo de su historia, los sistemas operativos, han adoptado principalmente alguna de las siguientes estructuras, bien como opción única o como alguna configuración mixta: estructura monolítica, estructura en capas y micronúcleo cliente/servidor.

Estructura monolítica

Antiguamente este era el tipo de organización más común. Este tipo de estructura consiste en que no existe estructura alguna, es decir, el sistema operativo está formado por **un único programa** con una colección de procedimientos que pueden llamarse entre sí sin ningún tipo de limitación cada vez que sea necesario. Estos sistemas tienen la ventaja de ser muy rápidos en su ejecución, solo hay que ejecutar un programa, pero cuentan con el inconveniente de carecer de la flexibilidad suficiente para soportar diferentes ambientes de trabajo o tipos de aplicaciones. Es por esto que estos sistemas operativos suelen ser hechos a medida, para solucionar un problema en concreto.

Estructura en capas

Se trata de la idea de construir sobre el hardware **niveles virtuales** superpuestos hasta llegar al nivel del usuario final. La organización más común de los sistemas actuales responde a la siguiente jerarquía:

- Capa 0: gestión del hardware del sistema informático.
- Capa 1: planificación del uso de la CPU.
- Capa 2: administración de la memoria.
- Capa 3: gestión de la E/S y de archivos.
- Capa 4: interfaz de llamadas al sistema.
- Capa 5: programas de usuario.

El concepto de **capa** o **anillo** está totalmente relacionado con el de **privilegio de ejecución**. Cuanto más cerca del nivel 0 se encuentre una capa, mayor privilegio asociado tendrán los procesos que en ella se ejecutan, así el máximo nivel corresponde a la capa 0, que teóricamente debería de ser la única capaz de dialogar directamente con el hardware del sistema.

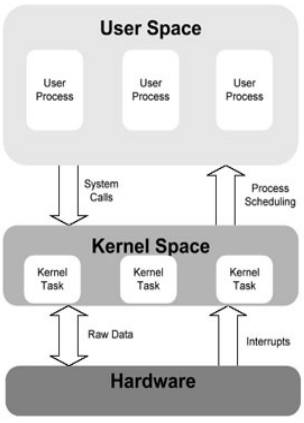
Micronúcleo y cliente/servidor

Los sistemas operativos modernos están evolucionando en la línea de desplazar la mayor cantidad posible de código a las capas más cercanas al nivel de usuario, con el fin de mantener un **núcleo de tamaño mínimo**, al que se le suele denominar micronúcleo o microkernel.

En este tipo de organización para solicitar un servicio al sistema, un proceso de usuario al que se denomina **proceso cliente**, envía una solicitud a un **proceso servidor**, que lleva a cabo el trabajo y devuelve la respuesta requerida.

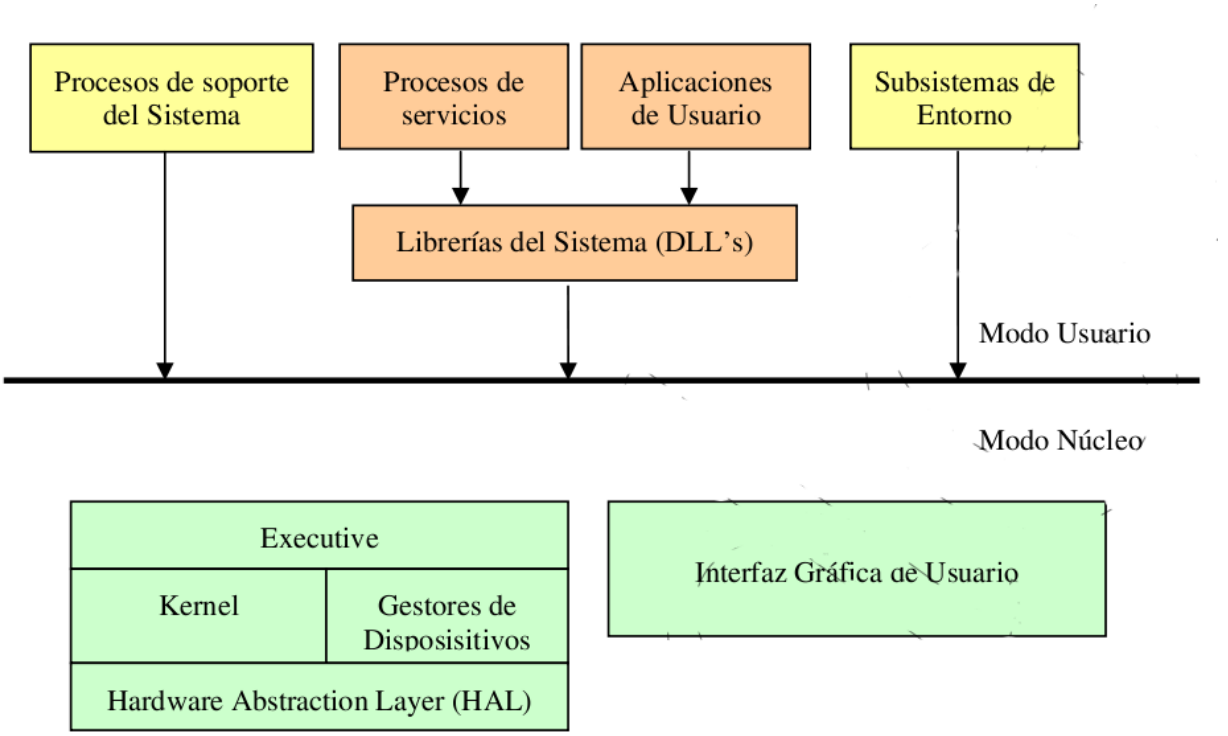
En este modelo, la función del núcleo es controlar la comunicación entre los procesos clientes y los servidores. Aunque en teoría la división del sistema operativo en micronúcleo y procesos clientes/servidores resulta una idea muy atractiva, no es directamente trasladable a la realidad. Algunas funciones del sistema operativo son de muy difícil realización en base a procesos que no tengan un nivel de privilegio máximo, esto se suele solucionar haciendo que algunos procesos críticos se ejecuten en realidad en modo núcleo, en lugar de modo usuario, con acceso total al hardware o estableciendo mecanismos para que estos procesos tengan canales de comunicación con el hardware lo más fluido que sea posible.

La comunicación entre los programas en el espacio de usuario y el espacio del kernel se realiza a través de las **llamadas al sistema**. Estas llamadas típicamente lo que hacen es acceder a recursos físicos compartidos. Todos los accesos a los recursos hardware son controlados por el kernel de modo que los programas de usuario no tienen que conocer los detalles físicos de los dispositivos.



Estructura Mixta

Como ejemplo de estructura mixta vamos a ver de manera simplificada la estructura de los **sistemas Windows** actuales, que combinan ideas de las estructuras anteriores.



Los componentes en modo núcleo de Windows se estructuran en un **modelo de capas**, donde encontramos los siguientes elementos:

- Executive: lo constituyen los servicios básicos del sistema operativo, lo que incluye gestión de memoria, gestión de hilos y procesos, seguridad, entrada/salida y comunicación entre procesos.
- Kernel: esta formado por las funciones de más bajo nivel, tales como planificación de hilos, atención de excepciones e interrupciones, y sincronización entre procesadores.
- Los Gestores de Dispositivos, que comprenden tanto a los gestores de dispositivos hardware de entrada/salida (traducen peticiones abstractas de entrada/salida a comandos específicos del dispositivo).
- El Hardware Abstraction Layer (HAL) es una capa de código que aísla al Kernel, a los gestores de dispositivos y al resto de Windows de detalles específicos del hardware.

Desde el punto de vista de que buena parte de la funcionalidad del núcleo se ha extraído de éste y se implementa como procesos externos al núcleo, se puede considerar un sistema operativo con **organización micronúcleo**.

Ahora bien, a diferencia del modelo micronúcleo puro en el que el núcleo gestiona solamente la multiprogramación, la gestión de interrupciones y la comunicación entre procesos, el núcleo de Windows incorpora en su propio espacio de memoria la gestión de dispositivos (lo que es propio de una **organización monolítica**). Esta decisión se ha tomado por cuestiones de eficiencia, ya que a efectos prácticos los gestores de dispositivos tienen un alto grado de dependencia con el núcleo y viceversa; permitir que esta interacción se efectúe mediante acceso directo es mucho más eficiente que efectuarlo mediante primitivas del sistema. Por supuesto, esto implica que un gestor de dispositivo (un driver, en terminología Windows) incorrectamente codificado puede corromper el núcleo y causar la caída del sistema. Los diseñadores de Windows se defienden de esta crítica argumentando que en un sistema con arquitectura micronúcleo pura en la que un gestor de dispositivo fallase, el sistema también terminaría cayendo al no funcionar correctamente el dispositivo gestionado.

6.2. Funciones de los Sistemas Operativos

Las funciones de los sistemas operativos son diversas y han ido evolucionando con el tiempo. Como principales funciones, se pueden enumerar las siguientes:

- **Gestión de procesos.** Hay que diferenciar entre los conceptos programa y proceso. Un programa es un ente pasivo, que cuando se carga en memoria y comienza a ejecutarse, puede originar una gran cantidad de procesos. El sistema operativo ha de cargar los distintos procesos, iniciarlos, supervisar su ejecución llevando a cabo los cambios de contexto necesarios y detectar su terminación normal o anormal.
- **Gestión de la memoria.** El sistema operativo debe administrar la memoria, es preciso separar adecuadamente las áreas a las que tiene acceso cada proceso, y repartir correctamente el siempre escaso recurso de la memoria principal.

- **Gestión de los dispositivos de E/S.** El acceso a los canales de E/S, y por extensión a los periféricos a ellos conectados, es una función muy importante que ha de llevar a cabo un sistema operativo. La gestión de la E/S tiene como objetivo proporcionar una interfaz de alto nivel sencilla de utilizar. En algunos sistemas operativos esta interfaz es semejante a la gestión de archivos.
- **Gestión de archivos.** El sistema de archivos, compuesto usualmente por directorios, subdirectorios y ficheros, es un componente principal de cualquier sistema informático. La gestión del espacio libre/ocupado, las asociaciones entre los bloques físicos de los dispositivos

Hay todo un conjunto de funciones adicionales, como la gestión general de errores del sistema, la gestión de la red, mecanismos de protección y seguridad, etc.. La lista de tareas de las que pueden hacerse cargo los sistemas operativos va incrementándose conforme pasa el tiempo y evoluciona la tecnología.

6.3. Sistemas Operativos actuales

Contar los usuarios de cada sistema operativo es una tarea imposible de realizar, en este apartado vamos a ver, como curiosidad, una aproximación del uso de algunos de los sistemas operativos más habituales.

Los informes periódicos realizados por la web *w3counter* pueden ser un indicador del uso de los sistemas operativos que acceden actualmente a Internet. El siguiente gráfico muestra el porcentaje de uso de cada sistema operativo basado en las visitas a webs diferentes.

Top 10 Platforms		
1	Android 7	16.15%
2	Windows 7	12.79%
3	Android 6	11.40%
4	iOS 11	11.36%
5	Windows 10	10.93%
6	Android 5	9.98%
7	Android 8	5.81%
8	Android 4	5.17%
9	Mac OS X	2.92%
10	Windows 8.1	2.16%

Otro dato curioso es la lista de los 500 ordenadores más grandes del mundo, que se publican el la web *"www.top500.org".* En una tendencia que se mantiene desde hace tiempo, más del 99% están basados en derivados de Unix, y más del 78% utilizan Linux.

A continuación, algunos números interesantes:

Núm.	Fabricante	Ordenador	Micros	S.O.
1	IBM	eServer Blue Gene	131072	CNK/Linux
2	IBM	eServer Blue Gene	40960	CNK/Linux
3	IBM	eServer pSeries p5 575 1.9 GHz	12208	AIX
4	SGI	SGI Altix 1.5 GHz	10160	Linux
5	Bull	NovaScale 5160, Itanium2 1.6 GHz	8704	Linux
6	Dell	PowerEdge 1850, 3.6 GHz	9024	Linux
7	NEC/Sun	Sun Fire X4600 Cluster, Opteron 2.4/2.6 GHz	10368	Linux
8	IBM	eServer Blue Gene	16384	CNK/Linux
9	Cray	Red Storm Cray XT3, 2.0 GHz	10880	UNICOS/Linux
10	NEC	Earth-Simulator	5120	Super-UX
...	...	...	...	...
130	Dell	PowerEdge 1855, 3.2 GHz	900	Windows Compute Cluster Server 2003

En la tabla se pueden ver las 20 primeras entradas de la lista, con el añadido del primer sistema no-Unix, que aparece en la posición 130. A continuación un resumen por tipos de sistemas operativos utilizados en estos 500 ordenadores:

Sistema Operativo	Cantidad	Porcentaje
Linux	367	73,40%
Basados en Unix	107	21,40%
Mixtos	24	4,80%
Windows	2	0,40%

7. VIRTUALIZACIÓN

La **virtualización** es una tecnología que permite simular por software recursos físicos. En un entorno virtualizado completo, cada sistema operativo tiene la ilusión de residir en una máquina individual con todos los recursos hardware disponibles enteramente para él.

Para conseguir una virtualización completa es necesario un programa, denominado **virtualizador** o **hipervisor**, según la técnica concreta que se utilice, que se encargue de gestionar el uso del hardware.

A cada una de las **máquinas virtuales** se les pueden asignar recursos hardware a través de distintas configuraciones. Entre los recursos que pueden ser compartidos encontramos: memoria RAM, CPU, disco duro, tarjetas de red, etc… En cada una de estas máquinas podemos instalar un sistema operativo y sus aplicaciones independientes. Así, por ejemplo, cuando un sistema operativo virtualizado cree que está escribiendo en un disco duro real, en realidad lo hace en un fichero, gestionado por la máquina virtual, que simula dicho disco duro.

7.1. Conceptos de virtualización completa

Los dos conceptos más importantes para entender qué es la virtualización completa son los de **anfitrión** e **invitado**.

El **anfitrión** es el sistema operativo del ordenador en el cual instalamos nuestro programa de virtualización y que asignará o prestará determinados recursos de hardware a la máquina virtual que creemos. Al software de virtualización se le llama **Hipervisor** o **Virtual Machine Manager** (VMM).

El **invitado** es el sistema operativo que instalamos en el ordenador virtual que hemos creado, mediante nuestro programa de virtualización y al cual hemos asignado determinados recursos para funcionar. A una instancia del hardware virtualizado se la conoce como **Máquina Virtual** (VM).

Es decir, el anfitrión (también conocido como host) es el que alberga al invitado (también conocido como guest). Un anfitrión puede tener varios invitados y los invitados no deben interferir entre ellos ni con el anfitrión. Los sistemas operativos invitados corren dentro de una Máquina Virtual.

Para construir la máquina virtual tenemos que asignar determinados recursos de hardware, como son espacio en disco duro, memoria RAM, número de procesadores, etc.. que el anfitrión cederá o compartirá con el invitado.

Los requisitos de hardware para un equipo que ejecute máquinas virtuales variarán en función de una serie de factores, entre los que se incluyen los siguientes:

- El software de virtualización instalado en el equipo anfitrión.
- El número y el tipo de sistemas operativos invitados.
- Las aplicaciones que tiene previsto ejecutar en los sistemas operativos invitados.
- Las aplicaciones que se tienen previsto ejecutar, simultáneamente con los sistemas invitados, en el equipo anfitrión.

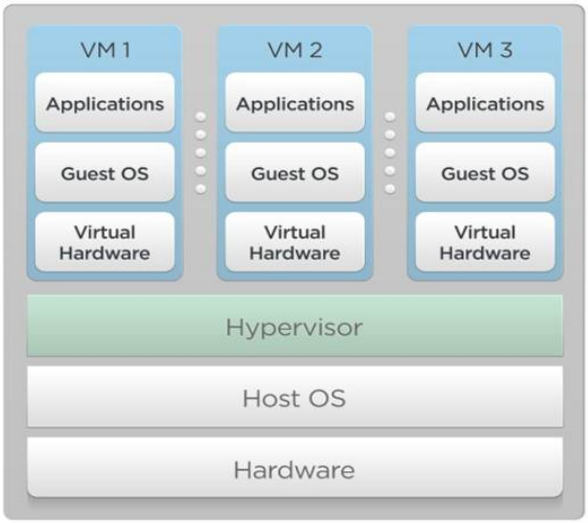
7.2. Técnicas de virtualización

Hay multitud de software que se puede utilizar para realizar virtualización y este software puede combinar una o varias técnicas diferentes, con lo cual se vuelve difícil una clasificación estricta de este software. A continuación se detallan algunas de las técnicas más comunes de virtualización.

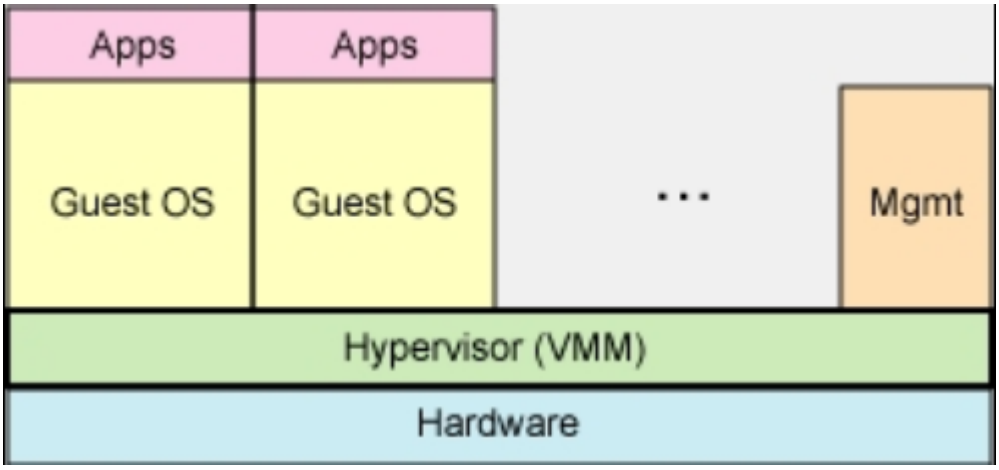
En la **virtualización completa** el recurso virtualizado es un sistema entero. En términos generales consiste en la abstracción de todo el hardware de una plataforma de manera que múltiples sistemas operativos puedan ejecutarse de manera independiente, con la ilusión de que los recursos les pertenecen en exclusiva. Esto es muy importante, ya que cada máquina virtual no ve a otra máquina virtual como tal, sino como otra máquina independiente de la que desconoce que comparte con ella ciertos recursos.

Algunos tipos de **virtualización completa** existentes son los siguientes:

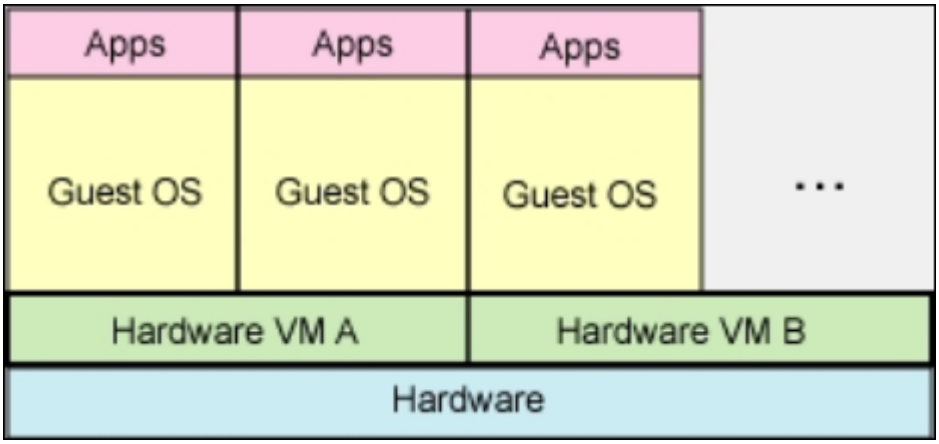
- **Hipervisor hosted.** El software de virtualización **se ejecuta sobre un sistema operativo**. La máquina virtual sólo simula el hardware necesario para permitir que un sistema operativo invitado se pueda ejecutar sobre el sistema operativo anfitrión. Algunos ejemplos de soluciones de este tipo son VMware Workstation, Parallels Desktop, Sun xVM, VirtualBox, VMware Player y Microsoft Virtual PC.



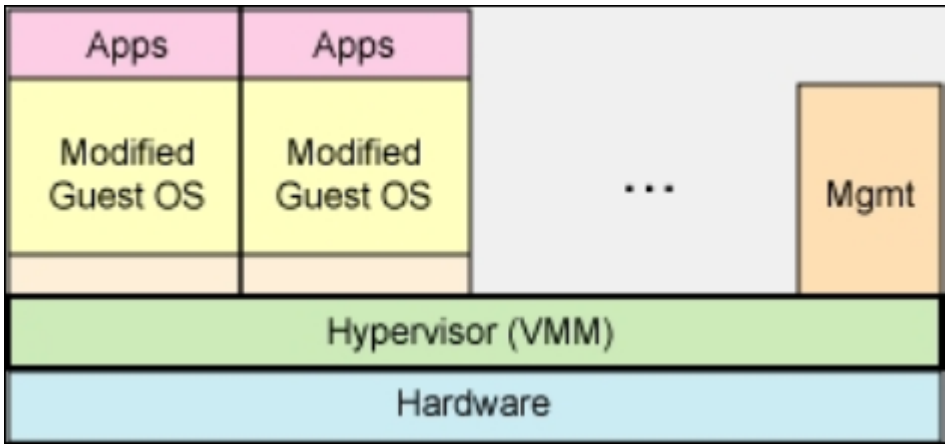
- **Hipervisor nativo, unhosted** o **baremetal**. La capa de virtualización se ejecuta **directamente sobre el hardware**, la cual incluye código que emula el hardware si es necesario, por lo que es posible ejecutar cualquier sistema operativo sin modificar. El código de emulación puede provocar pérdida en el rendimiento pero se puede hacer uso de soporte hardware específico para virtualización y así mejorar el rendimiento. Sin duda dentro de esta categoría podemos encontrar algunas de las soluciones más importantes sobre virtualización como VMware Server, XenServer, z/VM, Oracle VM, Sun xVM Server, Virtual Server, VMware ESX Server, VMware Fusion, Xen, Hyper-V.



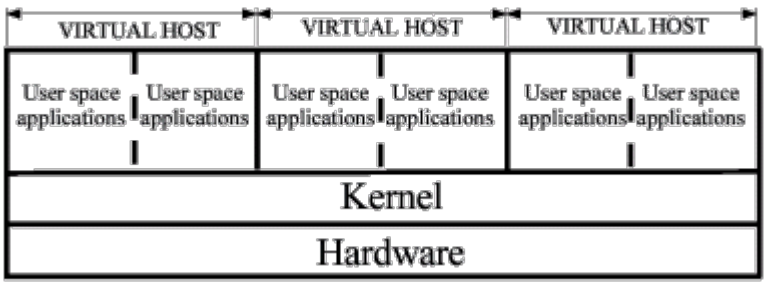
- **Emulación.** Un emulador que **replica una arquitectura hardware al completo** (procesador, juego de instrucciones, periféricos,..) y permite que se ejecuten sobre él máquinas virtuales. Por lo tanto se permite la ejecución de sistemas operativos y aplicaciones distintos al instalado físicamente en la máquina que ejecuta el emulador. Los emuladores más importantes actualmente son Bochs, MAME, DOSBox, Hercules, MESS, VirtualPC, y Qemu.



- **Paravirtualización.** Similar a la virtualización nativa porque introduce un hipervisor como capa de virtualización, pero además de no incluir emulación completa del hardware, **introduce modificaciones en los sistemas operativos invitados**, así éstos cooperan en la virtualización mejorando el rendimiento hasta obtenerlo casi similar a un sistema no virtualizado. Las soluciones más extendidas e importantes dentro del paradigma de la paravirtualización son : Xen, Logical Domains, Oracle VM, y Sun xVM Server.



- **Virtualización a nivel del sistema operativo** o **ligera.** Virtualiza sobre el propio sistema operativo, **sin introducir una capa intermedia** de virtualización. Por lo tanto, simplemente aísla las máquinas virtuales, que comparten el mismo sistema operativo. Aunque requiere cambios en el núcleo del sistema operativo, ofrece rendimientos próximos al sistema sin virtualizar. Compartiendo el mismo núcleo, entonces las máquinas no pueden correr sistemas operativos diferentes (sí distintas distribuciones Linux o versiones del sistema operativo dependiendo de la solución utilizada). Como ejemplos representativos de este modelo podemos citar OpenVZ, Linux V-Server, Virtuozzo, FreeBSD´s chroot jails, Free VPS, Solaris Containers y Solaris Zones.



En la **virtualización de recursos** se virtualiza únicamente un recurso individual de un ordenador, como puede ser la conexión a red, el almacenamiento principal y secundario, o la entrada y salida.

Existe un gran número de ejemplos dentro de la virtualización de recursos, como por ejemplo el uso de memoria virtual, los sistemas RAID (Redundant Array of Independent Disks), LVM (Logical Volume Manager), NAS (Network-Attached Storage) o la virtualización de red.

Algunos ejemplo de virtualización de recursos son:

- **Memoria virtual.** Permite hacer creer al sistema que dispone de mayor cantidad de memoria principal. Como sabemos, es usada en todos los sistemas operativos modernos. Por lo tanto, en este caso el recurso individual que es abstraído es la memoria y disco. Ejemplos conocidos por todos son el espacio Swap utilizados por los sistemas operativos Unix, o las técnicas de paginado de memoria usadas en sistemas operativos Microsoft.

- **Virtualización de almacenamiento.** Abstracción completa del almacenamiento lógico sobre el físico. Como ejemplos de virtualización de almacenamiento tenemos soluciones tan extendidas como RAID (Redundant Array of Independent Disks), LVM (Logical Volume Manager), SAN (Storage Area Network), NAS (Network-Attached Storage), NFS (Network File Systems), AFS, GFS, iSCSI (Internet SCSI), AoE (ATA over Ethernet).

- **Virtualización de red.** La virtualización de red consiste en la creación de un espacio de direcciones de red virtualizado dentro de otro o entre subredes. Es fácil ver que el recurso abstraído es la propia red. Ejemplos bien conocidos de virtualización de red son OpenVPN y OpenSwarm, que permiten crear VPNs.

En la **virtualización de aplicaciones,** las aplicaciones son ejecutadas encapsuladas sobre el sistema operativo de manera que aunque creen que interactúan con él y con el hardware de la manera habitual, en realidad no lo hacen, sino que lo hacen bien con una máquina virtual de aplicación o con algún software de virtualización. Este tipo de virtualización es usada para permitir a las aplicaciones ser ejecutadas en sistemas operativos para los cuales no fueron implementadas.

Debe quedar claro que la virtualización es solamente de las aplicaciones, lo que no incluye al sistema operativo anfitrión.

Un ejemplo bien conocido es wine, que permite la ejecución de aplicaciones de Microsoft Windows virtualizadas correr sobre GNU/Linux, dentro de lo que son llamadas técnicas de simulación. Otros ejemplos muy importantes son JVM (Java Virtual Machine, entorno de ejecución para lenguaje Java de Sun Microsystems) y CLR (Common Language Runtime, entorno de ejecución para la plataforma .NET de Microsoft).

Podemos diferenciar además entre los dos siguientes tipos de virtualización de aplicaciones:

- **Virtualización de aplicaciones limitada.** Aplicaciones Portables. Aplicaciones que pueden correr desde dispositivos de almacenamiento extraíbles. También se incluyen dentro de esta categoría las aplicaciones heredades que son ejecutadas como si lo hicieran en sus entornos originales. Lo normal es que en este caso, en virtualización de aplicaciones limitada, no medie ninguna capa de virtualización o software con las mismas prestaciones y que la portabilidad se encuentre limitada al sistema operativo sobre el que correrá la aplicación. El recurso abstraído es el sistema operativo sobre el que son ejecutadas las aplicaciones virtualizadas.
- **Virtualización de aplicaciones completa.** En este segundo tipo de virtualización de aplicaciones, una capa intermedia o software de virtualización es introducido para mediar entre la aplicación virtualizada y el sistema operativo y hardware.
  - **Portabilidad Multiplataforma** (Cross-platform). Permite a aplicaciones compiladas para una CPU y sistema operativo específicos ser ejecutadas en diferentes CPUs y sistemas operativos sin ser modificadas, usando una traducción binaria dinámica y mapeado de llamadas del sistema operativo. No requiere recompilación o porting al correr en un entorno virtualizado,

normalmente una máquina virtual de proceso o aplicación. Por tanto, el recurso abstraído en este caso es la CPU y el sistema operativo. Ejemplos utilizados en la mayoría de los sistemas son Java Virtual Machine, Common Language Runtime, Mono, LLVM, Portable .NET, Perl Virtual Machine, Citrix XenApp, Novell ZENworks Application Virtualizacion, VMware ThinApp, Microsoft Application Virtualization.

- **Simulación.** Reproducción del comportamiento de una aplicación concreta o una funcionalidad específica de una aplicación. Ahora, el recurso que se abstrae es la API (Application Program Interfaces) del sistema operativo, o cualquier interfaz. Antes ya se comentó Wine como ejemplo de este modelo de virtualización de aplicaciones, además disponemos de Crossover office, coLinux, Zebra, o Quagga.

7.3. Ventajas y desventajas de la virtualización

Hemos visto algunos conceptos de la virtualización, vamos a hablar ahora de las ventajas de la virtualización, y por qué se ha producido en los últimos años el despegue definitivo de este tipo de tecnología. Aunque ya hacía bastante tiempo que la virtualización se usaba, ahora se ha producido la implantación definitiva en muchas empresas, sobre todo de gran tamaño. Y la adopción por gran parte de las empresas se debe sin lugar a dudas a las ventajas que este tipo de tecnología ofrece como puede ser:

- Ahorro de costes: es una de las cuestiones por las cuales más se han interesado las empresas en la virtualización, puesto que donde antes necesitaban dos máquinas ahora se puede utilizar sólo una. Pero no sólo queda aquí, sino que además podemos ahorrar mucho tiempo gracias a la facilidad de administración o de clonación de los discos duros virtuales, que se realizarán como cualquier otro archivo, con las ventajas que esto tiene asociado.
- Entornos de prueba: para probar versiones no definitivas de un programa (versiones beta) interesa virtualizar nuestro propio sistema para realizar todas estas instalaciones en el sistema virtual y dejar nuestro sistema anfitrión “limpio”, instalando sólo aquello que definitivamente vamos a usar.

- Entornos aislados de seguridad: crear un sistema aislado donde las únicas conexiones con internet se harán en entornos seguros.
- Compatibilidad de programas: cuando se utiliza un sistema operativo y queremos usar o probar algún programa diseñado para otro sistema operativo.
- Rápida incorporación de nuevos recursos para los servidores virtualizados, es muy fácil asignar hardware nuevo a una máquina virtual.
- Reducción de los costes de espacio y consumo necesario.
- Administración global centralizada y simplificada.
- Aislamiento, un fallo general de sistema de una máquina virtual no afecta al resto de máquinas virtuales. Un fallo en la maquina física se solventa sustituyendo la maquina completa, las máquinas virtuales se pueden montar en cuestión de minutos sobre esta nueva máquina, por lo que la empresa no queda paralizada mientras se procede a la instalación manual de todos los sistemas.
- Reduce los tiempos de parada necesarios. Migración en caliente de máquinas virtuales (sin pérdida de servicio) de un servidor físico a otro, eliminando la necesidad de paradas planificadas por mantenimiento de los servidores físicos.
- Balanceo dinámico de máquinas virtuales entre los servidores físicos que componen el pool de recursos, garantizando que cada máquina virtual ejecute en el servidor físico más adecuado y proporcionando un consumo de recursos homogéneo y óptimo en toda la infraestructura.
- Posibilidad de migración de toda nuestra infraestructura de una localización a otra de forma muy simple.
- Simplificación de la administración de sistemas, posibilidad de que el administrador cuente con toda la infraestructura clonada para fines de prueba y aprendizaje.

Como desventajas podemos citar:

- Muchos sistemas dependen de un solo equipo, siendo necesario aplicar medidas extra para asegurar la integridad de los datos.
- El rendimiento de los sistemas se reduce ya que se comparten recursos físicos.
- Pueden aparecer problemas en la compatibilidad con el hardware virtualizado.

7.4. Solución de virtualización VirtualBox

Una de las plataformas más interesantes a la hora de probar la virtualización es **VirtualBox**, la plataforma de virtualización de Sun Microsystems (ahora propiedad de Oracle Corporation), es un programa para facilitar la virtualización completa de sistemas operativos que podremos instalar tanto en Windows, Mac y GNU/Linux.

La sencillez es una de las mayores ventajas para el usuario que quiera comenzar a conocer la virtualización. VirtualBox es un **hipervisor hosted** de virtualización completa que cuenta con 2 versiones:

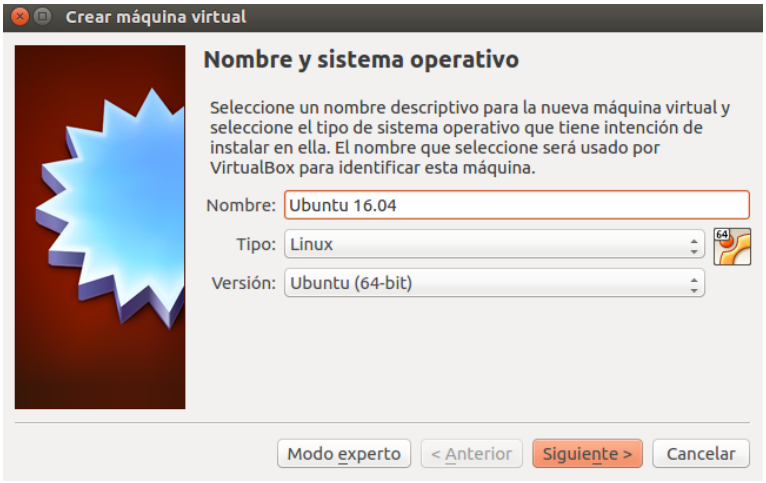
- **Virtualbox OSE** (Open Source Edition): es la versión software libre y tiene licencia GPL y se puede encontrar en el repositorio community. Le faltan algunas características como la capacidad para usar dispositivos USB y un servidor RDP.
- **Oracle VM VirtualBox:** tiene licencia privativa y sólo es gratuita para evaluación y para uso personal (VirtualBox Personal Use and Evaluation License o PUEL). Ésta es la versión completa de VirtualBox con todas sus características y se puede encontrar en los repositorios oficiales, o se puede bajar de la sección de descargas en el sitio web de VirtualBox www.virtualbox.org/wiki/Downloads

A continuación se detalla el uso básico de VirtualBox pudiendo consultar los manuales oficiales (en inglés) para realizar un uso más avanzado de la aplicación: www.virtualbox.org/wiki/Documentation

Una vez descargado lo instalamos con el procedimiento habitual según el sistema operativo que estemos usando, una vez iniciado veremos la pantalla de inicio, de la cual nos interesa la parte superior donde está la opción de crear una nueva **máquina virtual**.

Al hacer clic en “Nueva” vamos a poder decirle qué tipo de sistema operativo vamos a instalar en esa **máquina virtual**.

En esta parte podemos **ponerle un nombre a nuestra máquina virtual** para diferenciarla del resto que tengamos, además de elegir **el tipo y la versión** de sistema operativo huésped.



Ahora debemos elegir cuánta **cantidad de memoria RAM** tendrá la máquina que estamos creando. El valor que aparece por defecto debería ser suficiente, aunque la mejor opción es ajustar la memoria a los requisitos recomendados por cada sistema operativo. Teniendo siempre en cuenta que la memoria que asignemos a la máquina virtual no estará disponible para la máquina real y resto de máquinas virtuales cuando estén en ejecución. Una vez elegida la cantidad de memoria de la que dispondrá el ‘nuevo ordenador’ hacemos clic en “Siguiente”.

Aquí elegimos que queremos “crear un disco duro virtual ahora” y le damos a “*Crear*”.

Si no tenemos intención de compartir esta máquina con nadie que use un software de virtualización distinto a **VirtualBox** podemos seleccionar el tipo de archivo de disco duro como **VDI** (VirtualBox Disk Image).

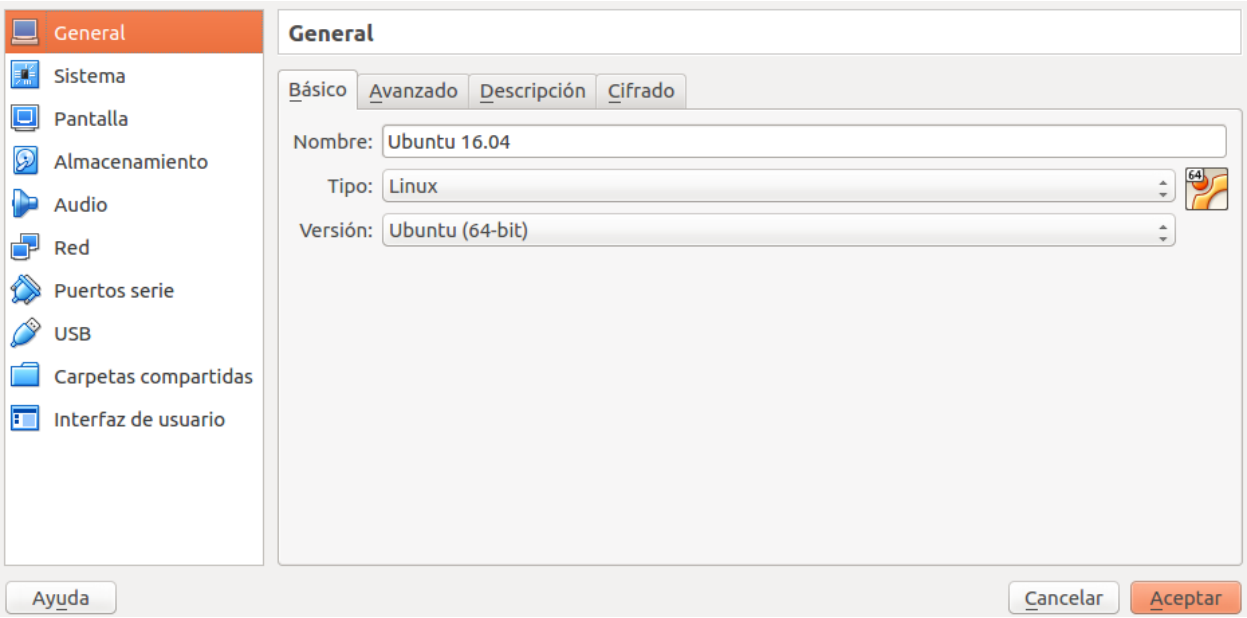
Ahora tenemos que elegir qué **tipo de disco duro** queremos agregarle a la máquina: uno que tenga un tamaño fijo desde el principio o uno que tenga un tamaño pequeño al principio y vaya aumentando con el tiempo y el uso. Aquí hay que hacer una distinción sobre cuál elegir:

- **Reservado dinámicamente**: de esta forma el archivo que simula el disco duro del ordenador tendrá un tamaño pequeño al principio e irá aumentando cuando vayamos llenando el disco duro virtual. Tiene la ventaja de que es más rápido al crear ahora el disco duro y que inicialmente ocupa menos espacio en el disco duro real, pero hace que la máquina virtual sea menos eficiente.

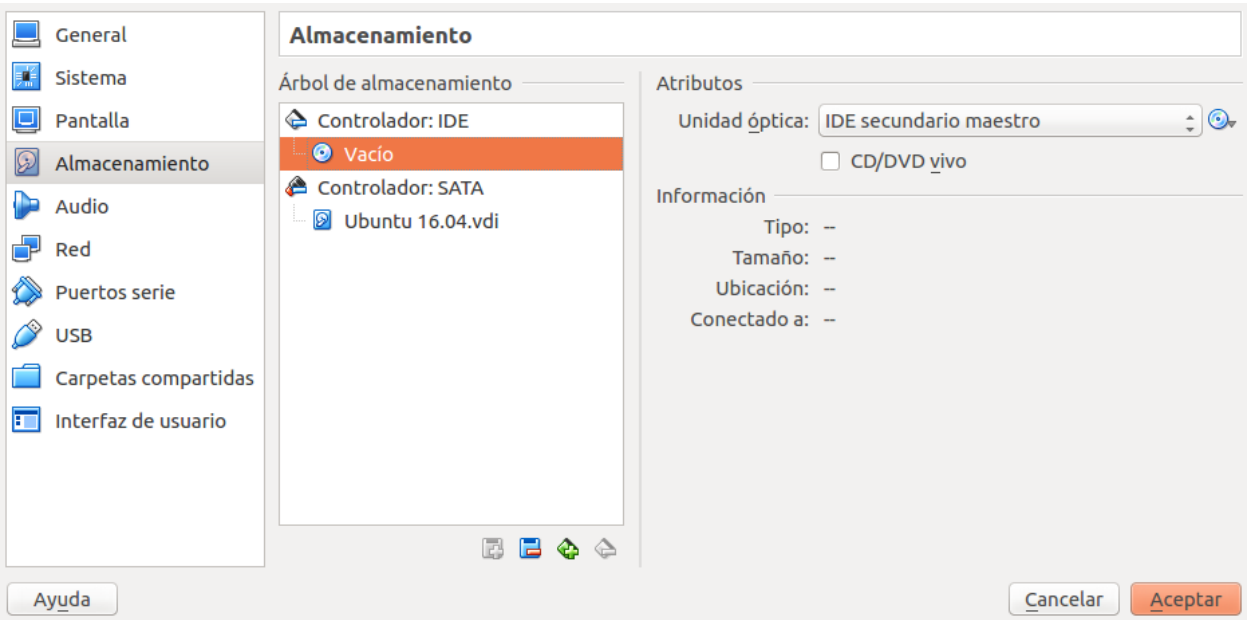
- **Tamaño fijo**: en este caso, se crea el disco duro completo en el momento de la creación de la máquina virtual, lo que hace el proceso de creación un poco más largo ya que crea un fichero del tamaño del disco duro que seleccionemos, pero repercute de forma positiva a la hora de usar el sistema operativo huésped.

Elegimos el **nombre y la ubicación de nuestro disco duro y la cantidad de espacio que tendrá**. Normalmente con el tamaño predeterminado es suficiente, aunque dependerá del uso que le vayamos a dar a la máquina virtual. Este tamaño es el tamaño del disco duro que verá que tiene disponible la máquina virtual y será el tamaño del fichero que virtualiza el disco duro en nuestra máquina real, si hemos seleccionado disco de tamaño fijo, o el tamaño máximo que podrá llegar a tener si hemos seleccionado reservado dinámicamente.

Ya tenemos nuestra **máquina virtual** creada y disponible para iniciarla con la memoria RAM y el disco duro que hemos configurado. Pero antes debemos comprobar y configurar otros recursos. Para ello con la máquina virtual seleccionada nos vamos a **"Configuración"** y veremos una pantalla como la siguiente:



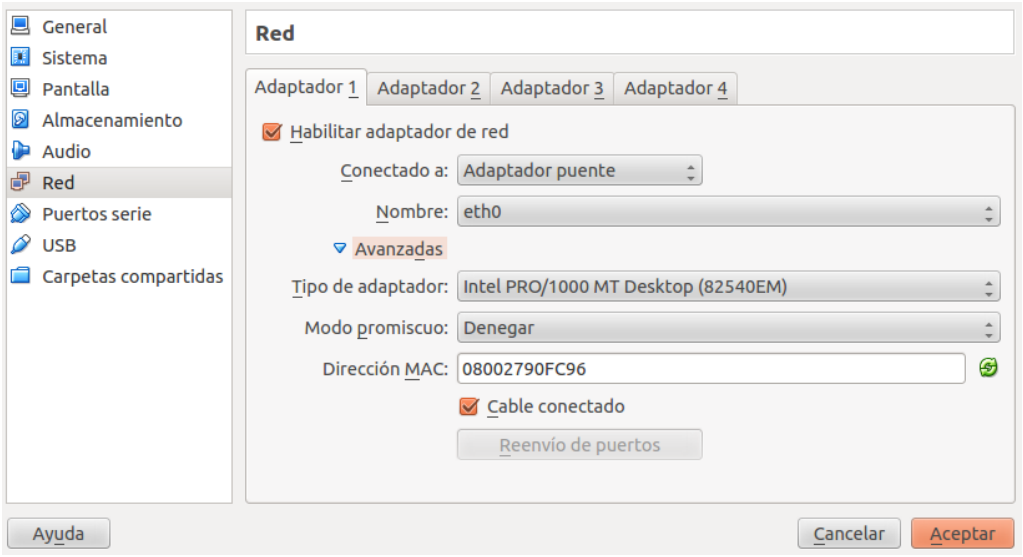
Una de las cosas básicas a configurar para comenzar la instalación del sistema operativo es en **"*Almacenamiento*"** donde seleccionamos la unidad de CD/DVD como se ve en la imagen.



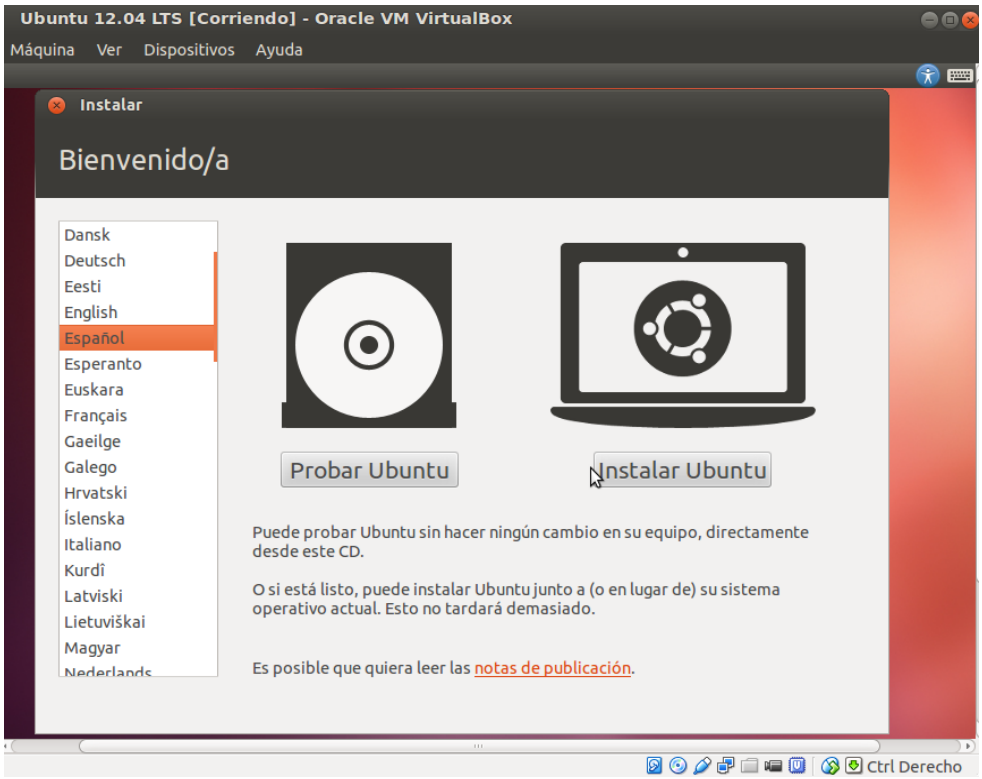
Una vez ahí, podemos indicarle a VirtualBox si queremos que use el CD/DVD de la máquina real o que monte una imagen de un CD/DVD grabado, si queremos que monte la imagen de un CD/DVD le damos al icono del CD que aparece en la parte derecha de la pantalla y elegimos "*Seleccionar un archivo de disco virtual de CD/DVD*", elegimos la imagen del CD de instalación y "*Abrir*". Después de haber agregado el CD, le damos a "*Aceptar*" y ya tendremos nuestra máquina **lista para el primer arranque**.

Antes de iniciar la instalación de la máquina virtual, otro apartado interesante para configurar es el de **Red.** En este apartado podemos habilitar hasta 4 adaptadores de red (tarjetas de red). Y los podemos configurar desde "Conectado a" como:

- No conectado. VirtualBox muestra un adaptador de red pero sin conexión. (cable desconectado)
- NAT (Network Address Translation). Permite funcionalidad básica desde el sistema operativo huésped, navegar por internet, acceder al correo, descargar ficheros. Pero tiene bastantes limitaciones si tenemos que establecer conexiones con la máquina virtual ya que la IP de la máquina virtual es transformada por la de la máquina real cuando accede a la red.
- Red NAT. Permite hacer uso de una red configurada desde "Archivo → Preferencias → Red". Simula un router que puede servir la configuración de red por DHCP a las máquinas virtuales en una red definida.
- Adaptador puente. Simula una conexión física real a la red, asignando una IP al sistema operativo huésped. Esta IP se puede obtener por DHCP o directamente configurándola en el sistema operativo huésped. Es decir, la máquina virtual es un equipo más de la red.
- Red interna. Comunicación restringida entre las maquinas virtuales conectadas en la misma red interna. Esta limitación viene justificada por seguridad y velocidad.
- Adaptador sólo-anfitrión. Se hace una simulación de un adaptador lógico de red, similar a la interface de loopback, solo permite la comunicación internamente en la maquina virtual.
- Controlador genérico: permite configurar un adaptador de red con sus propios drivers a través de un paquete que se descarga por separado.



Una vez configurados los aspectos básicos de la máquina virtual podemos arrancarla, para ello sólo tenemos que seleccionarla en la lista y darle al botón **"*Iniciar*".**



8. SEGURIDAD INFORMÁTICA

8.1. Introducción a la seguridad

Por **seguridad de la información** entendemos el conjunto de medidas y procedimientos, tanto humanos como técnicos, que permiten proteger la integridad, confidencialidad y disponibilidad de la información. Este término, por tanto, es un concepto amplio que engloba medidas de seguridad que afectan a la información independientemente del tipo de ésta, soporte en el que se almacene, forma en que se transmita, etc.

La **seguridad informática**, por su parte, es una rama de la seguridad de la información que trata de proteger la información que utiliza una infraestructura informática y de telecomunicaciones para ser almacenada, procesada o transmitida.

Por tanto, podemos decir que la seguridad de la información es un concepto mucho más amplio, que engloba a la seguridad informática. O dicho de otra forma, la seguridad de la información se encarga de proteger el sistema de información de una organización y la seguridad informática, se encarga de proteger los sistemas informáticos, que son el soporte o infraestructura de ese sistema de información.

La organización ISO/IEC, en su norma 27000 define lo que son los tres **principios básicos de la seguridad de la información**, o lo que se conoce también como triada CIA. Además de estos tres principios básicos que son la **confidencialidad**, **integridad** y **disponibilidad**, existen otros adicionales como son:

- Autenticación: consiste en verificar la identidad del usuario de la información como la de su creador.
- Control de acceso: permite restringir los accesos a los recursos en función de los permisos asignados a cada usuario.
- No repudio: relacionado con la autenticación, prueba la participación de ambas partes en una comunicación o transacción de tal forma que no se puede negar haber participado en ella. Puede ser:
  - En origen: el emisor no puede negar el envío, por ejemplo presentación telemática del IRPF.

    - En destino: el receptor no puede negar haber recibido la información pues el emisor tiene pruebas de la recepción
- Trazabilidad: es la capacidad de registro de las operaciones de un sistema informático, de manera que cualquier operación pueda ser rastreada hasta su origen. Nos va a permitir poder realizar un análisis forense de un incidente de seguridad.

En el campo de la seguridad de la información, se trabaja con una serie de términos y conceptos que conviene conocer. Destacamos los más importantes:

- Activo: recurso del sistema de información o relacionado con éste, necesario para que la organización funcione correctamente y alcance los objetivos propuestos.
- Amenaza: es un evento que puede desencadenar un incidente en la organización, produciendo daños materiales o pérdidas inmateriales en sus activos.
- Impacto: mide la consecuencia al materializarse una amenaza.
- Riesgo: mide la probabilidad de que produzca un incidente de seguridad en un activo, en un dominio o en toda la organización. El riesgo se relaciona con el impacto. El impacto mide lo que puede pasar. El riesgo lo que probablemente pase.
- Vulnerabilidad: debilidad inherente en cualquier sistema informático de ocurrencia de la materialización de una amenaza sobre un Activo. La vulnerabilidad se relaciona con la amenaza ya que un sistema puede ser más o menos vulnerable respecto a una amenaza.
- Ataque: evento, exitoso o no, que atenta sobre el buen funcionamiento del sistema.
- Desastre o Contingencia: interrupción de la capacidad de acceso a información y procesamiento de la misma a través de computadoras necesarias para la operación normal de un negocio.

En las empresas y organizaciones concienciadas con la seguridad de la información, antes de implantar un sistema de gestión de la seguridad de la información (SGSI), hay que realizar un análisis de riesgos e impactos previo.

Una amenaza es un evento que puede desencadenar un incidente en la organización a consecuencia de una vulnerabilidad existente. Las vulnerabilidades suelen ocurrir por causas:

- Tecnológicas, debido a fallos inherentes en la tecnología.
- De configuración, debido a que los dispositivos, servidores, aplicaciones, etc., vienen con configuraciones inseguras por defecto o no han sido bien configurados por el administrador.
- Políticas de seguridad, porque la organización carece de ninguna política de seguridad o bien porque no son correctas o no se han actualizado con las nuevas amenazas que aparecen.

Los tipos de amenazas pueden ser:

- Físicas: Las amenazas físicas y ambientales afectan a la parte física del sistema de información: instalaciones, hardware, control de acceso, etc. Son el primer nivel de seguridad a proteger.
- Lógicas: Las amenazas lógicas afectan a la parte lógica: sistemas operativos (S.O.'s), aplicaciones y datos.
- Pasivas: Las amenazas pasivas o escuchas, suponen un intento de un atacante para obtener información relativa a una comunicación, por ejemplo, capturar datos con un analizador de redes o sniffer como wireshark.
- Activas: Las amenazas activas son más peligrosas y su objetivo es la modificación de los datos transmitidos o la creación de trasmisiones falsas, por ejemplo, un ataque Man In The Middle (hombre en el medio, MITM) como el que podemos sufrir al conectarnos a una wifi abierta en un estación o aeropuerto. En un ataque MITM, el atacante generalmente suplanta al router de forma que el tráfico de la víctima pasa por el ordenador del atacante, pudiendo robar credenciales de acceso a servicios, como redes sociales o banca electrónica.

Detrás de muchas de estas amenazas, obviamente excepto los desastres naturales, se encuentra toda una taxonomía de atacantes, con fines lucrativos, de chantaje, activistas o simplemente de ego personal por el impacto del ataque realizado. Algunos de ellos son:

- Hacker (White hat)
- Cracker (Black Hat)
- Grey Hat
- Lamer y Script Kiddies
- Phreaker
- Spammer

- Phisher
- Scammer
- Ciberterroristas

Hay que destacar que en los medios, siempre se utiliza el término hacker de forma despectiva, incluyendo en este concepto a todos los ciberdelincuentes con fines maliciosos y que realmente encajan dentro de la definición de cracker. Existe una iniciativa dentro del propio movimiento hacker para cambiar esta definición por una más correcta en que se defina al hacker como persona experta en alguna o varias ramas de la tecnología informática y electrónica (redes, programación, sistemas operativos, dispositivos móviles, etc) que se dedica a intervenir y/o realizar alteraciones técnicas (to hack, del inglés) sobre un producto o dispositivo.

La ISO/IEC 27001 define la seguridad como un proceso de mejora continua. La seguridad informática es un proceso que comienza en el momento en que se implanta en la organización, pero nunca acaba, ya que lo que hagamos hoy puede que no sirva para mañana. La tecnología avanza muy rápidamente y de la misma forma que aparecen nuevas funcionalidades y maneras de hacer las cosas, surgen como consecuencia nuevos vectores de ataques. Por tanto, si somos los responsables de seguridad de una organización, debemos estar continuamente aplicando las fases del ciclo de vida de la seguridad:

También se le conoce como el modelo Plan-Do-Check-Act (PDCA o ciclo Demming) y que consiste en las siguientes fases:

- Planificación (Plan): Establecer la política, objetivos, procesos y procedimientos relativos a la gestión del riesgo y mejorar la seguridad de la información de la organización para ofrecer resultados de acuerdo con las políticas y objetivos generales de la organización.
  - Identificar lo que se quiere mejorar.
  - Recopilar datos del proceso que se quiere mejorar.
  - Analizar los datos recogidos.
  - Establecer los objetivos de mejora.
  - Detallar los resultados esperados.
  - Definir los procesos necesarios conseguir los objetivos.
- Ejecución (Do): Implementar y gestionar el Sistema de Gestión de la Seguridad de la Información (SGSI) de acuerdo a su política, controles, procesos y procedimientos. En la medida de lo posible debería hacerse en un entorno de prueba para poder verificar sus resultados antes de implantarlo en el sistema real.
- Seguimiento (Check): Verificar, medir y revisar las prestaciones de los procesos del SGSI. Comprobar que las medidas adoptadas han surtido efecto, para ello se debe volver a recopilar datos y monitorizar el comportamiento del sistema.
- Mejora (Act): Adoptar acciones correctivas y preventivas basadas en auditorías y revisiones internas con el objetivo de mejorar el SGSI. Hace referencia a la actitud que se debe tomar después de los tres primeros pasos y dependerá de lo que haya ocurrido. En caso de haber ocurrido algún mal funcionamiento, se deberá repetir el ciclo de nuevo. Si el funcionamiento ha sido correcto, se instalarán las modificaciones en el sistema de manera definitiva.

La seguridad debe considerarse en toda organización como una inversión y no como un gasto. Desgraciadamente no muchas organizaciones se han dado cuenta de eso y es fácil ver hoy en día muchas empresas sin políticas de seguridad, sin planes de contingencia o continuidad de negocio ante desastres o incumpliendo la Ley Orgánica de Protección de Datos.

Muchas empresa reaccionan cuando han sufrido un incidente de seguridad y no tienen planes de acción de forma que lo dejan todo a la improvisación. No tener un SGSI implantado puede llevar a ocasionar muchas pérdidas económicas e incluso la reputación y la imagen de la empresa ante un incidente de seguridad con gran impacto y que pueda ser anunciado en los medios de comunicación, dando una mala publicidad a la empresa.

También es verdad es que empieza a haber un cambio de tendencia en el sector y las organizaciones empiezan a demandar profesionales en el campo de la seguridad informática.

Para finalizar, las organizaciones deben plantearse muy seriamente la necesidad de formación y concienciación de sus empleados en materia de seguridad informática, ya que el factor humano es muy importante a la hora de evitar incidentes. Muchos de los problemas de derivan por malas prácticas en los empleados y por desconocimiento de los peligros que acechan en Internet, donde la mayoría de ataques se realizan mediante phishing o ingeniería social.

8.2. Seguridad física

Las medidas de seguridad física tratan de proteger los **activos tangibles y físicos** de la organización, así como a las personas. Así pues, en la seguridad física podemos tener medidas para protegernos o minimizar el impacto ante las siguientes amenazas:

- Incendios: Mobiliario ignífugo, muros y paredes cortafuegos, extintores…
- Inundaciones: Evitar ubicación en plantas bajas, impermeabilizar paredes y techos…
- Robos: Cámaras de seguridad, vigilantes, códigos de seguridad…
- Señales electromagnéticas: Evitar ubicación en lugar con gran radiación, proteger de las emisiones mediante filtros o cableado especial…
- Apagones: SAI, estabilizadores, grupos electrógenos.
- Sobrecargas eléctricas: SAI profesionales con filtros para evitar picos de tensión.
- Desastres naturales: Contactar con la Agencia Estatal de Meteorología, CPD's de respaldo...

Además, en las medidas de seguridad física y sobre todo, en las grandes organizaciones, entrarían todas las medidas de diseño, ubicación y acondicionamiento de lo que se conoce como Centros de Procesamiento de Datos (CPD), Centros de Cálculo o Centros de Datos (Data Centers), que son salas o edificios donde se ubican todos recursos físicos, lógicos y humanos necesarios para la organización, realización y control de las actividades informáticas de una empresa.

En cuanto a la protección de los sistemas informáticos es conveniente disponer de **sistemas de alimentación ininterrumpida** (SAI o UPS) para proteger los equipos informáticos críticos, como pueden ser servidores, ordenadores con información crítica o dispositivos de red como enrutadores o conmutadores, que en muchas ocasiones se han visto afectados por una subida de la tensión eléctrica generada por una tormenta.

El SAI o UPS mejora la calidad del suministro eléctrico minimizando problemas como:

- Pequeños cortes de energía
- Sobretensión o disminuciones de tensión
- Ruido eléctrico
- Distorsión armónica

La potencia del SAI se mide en VA (voltiamperios) generalmente, aunque también es habitual en W (watios) y hay que seleccionar un SAI que permita soportar la carga (número de equipos informáticos) que le vamos a conectar. Por ejemplo, un SAI de 1500W, no puede soportar teóricamente la conexión de 4 servidores si cada uno de ellos tiene una potencia de 500W en su fuente de alimentación, suponiendo que está funcionando a plena potencia. Para estos cálculos siempre es conveniente irse al peor caso y además, debido a posibles picos de consumo de los aparatos conectados, se recomienda siempre elegir un SAI con una capacidad de suministro un 20% mayor que el consumo que vamos a proteger.

También hay que indicar que la función de un SAI no es proporcionar alimentación continua cuando cae el suministro, sino permitir un apagado correcto de los equipos cuando queda poca batería. Para proporcionar energía permanente se usan generadores o grupos electrógenos, pero son más habituales en instalaciones críticas como hospitales, data centers, etc.

Los SAI suelen llevar un agente software que hay que instalar en los servidores que se conectan al SAI y que se comunican con el SAI por puerto COM o USB, avisando por red del nivel de batería y estado del SAI al resto de servidores. Generalmente un servidor hace de maestro estando conectado al SAI por puerto COM/USB y avisa al resto de agentes por red para que se apaguen si queda poca batería, normalmente en un umbral definido por el administrador cuando instala y configura el agente.

Existen varios tipos de SAI:

- Standby (offline): En funcionamiento normal, la carga crítica se alimenta directamente de la tensión de la red. Cuando hay un fallo de suministro, éste pasa a funcionar con sus baterías. Se usa para entornos domésticos. Precio bajo.
- Interactivo: Proporciona alimentación de su batería en cuanto la energía de la red baja, haciendo funciones de regulador de voltaje (AVR). Tiene un tiempo de conmutación a batería menor que los offline. Precio medio.
- On-line de doble conversión: Continuamente alimenta a los equipos conectados mediante la tensión que proporciona el inversor, totalmente aislada de la tensión de la red. Se llama de doble conversión porque convierte de corriente alterna (AC) a corriente continua (CC) y luego a corriente alterna de nuevo. Son los que se usan habitualmente en los CPD y son los más caros.

8.3. Seguridad lógica

La seguridad lógica protege los **activos intangibles** de la organización, como son el software, los datos o los procesos, así como de los accesos no autorizados a los sistemas informáticos. Por tanto, en la seguridad lógica podemos tener medidas para protegernos o minimizar el impacto ante las siguientes amenazas:

- Robos: Cifrar la información almacenada, utilizar contraseñas, sistemas biométricos…
- Pérdida de información: Copias de seguridad, sistemas tolerantes a fallos, discos redundantes RAID…
- Pérdida de integridad: Programas de chequeo del equipo, firma digital, herramientas de integridad…

- Software malicioso: Antimalware (antivirus, antirootkit, etc.)
- Ataques red: Firewall, programas de monitorización, servidores proxys, detectores de intrusos…
- Accesos no autorizados: Contraseñas, listas de control de acceso, cifrar documentos...

La seguridad lógica es la rama más importante de la seguridad informática porque es la que se centra en proteger principalmente los datos y por tanto, la información que es el activo más importante de una organización.

El cifrado de archivos de datos, nos permite garantizar la confidencialidad de los mismos, ya que sólo van a poder acceder a los datos aquellos usuarios que puedan descifrar el mensaje usando el mismo algoritmo de cifrado y la clave de descifrado.

La mejor forma de garantizar la confidencialidad de la información de nuestro disco para evitar que la información caiga en manos ajenas, es usar cifrado en el disco. Si alguien accede a nuestro ordenador físicamente o a nuestro disco, aunque hayamos creado permisos de acceso en los datos, cualquiera puede arrancar otro sistema operativo con un CD, DVD o USB y acceder con permisos de administrador o root, saltándose la barrera de permisos que hayamos creado.

Algunos sistemas operativos incluyen de serie herramientas para cifrar los contenidos del disco duro, en su sistema de archivos. Tal es el caso de Windows con su EFS y BitLocker, dos sistemas integrados en Windows con características diferentes. Los sistemas GNU/Linux también permiten cifrar el sistema de archivos y se puede indicar en la instalación del sistema operativo.

El **cifrado simétrico** de archivos nos va a permitir proteger el contenido de un archivo confidencial que pueda necesitar enviar a través de un medio inseguro, por ejemplo como adjunto a un correo electrónico. De esta forma, el destinatario del archivo puede descifrar el fichero usando la clave simétrica que previamente debe haberse acordado, utilizando el mismo algoritmo con el que se ha cifrado (como por ejemplo 3DES, BlueFish, IDEA, AES, etc)

A mediados de los años 70, la criptografía de clave pública, también conocida como de **clave asimétrica** viene a complementar la criptografía tradicional de cifrado simétrico y algunas de sus debilidades. Se conoce con este nombre porque la clave de cifrado es diferente a la de descifrado de forma que lo que se hace una clave, se deshace con la otra.

Cada participante en una comunicación, ya sea una persona o un sistema informático como un servidor web, utiliza un par de claves:

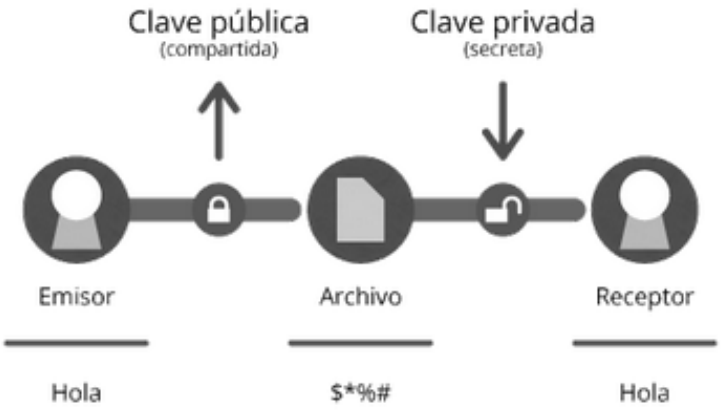
- Clave pública: se puede entregar o compartir con cualquier persona, ya que no es secreta. Incluso se puede publicar en páginas personales.
- Clave privada: el propietario debe guardarla a buen recaudo ya que no debe compartirse con nadie y como su nombre indica, es para uso privado de su poseedor.

La seguridad del sistema se basa por tanto en la seguridad de la clave privada y en que es computacionalmente "imposible", obtener la clave privada de una persona o servicio conociendo su pública. Cuando se habla de computacionalmente imposible, queremos decir que con la potencia actual de los ordenadores, se podría invertir decenas o cientos de años, en función de la fortaleza del algoritmo y de la longitud de las claves, en poder descubrir la clave privada.

El par de claves se generan mediante propiedades matemáticas de los números primos muy grandes. Esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

Para cifrar a un destinatario, el remitente cifra el mensaje con clave pública del destinatario de forma que sólo el destinatario puede descifrar con su privada, pues es el único que la posee. De esta manera se logra la confidencialidad del envío del mensaje, ya que nadie salvo el destinatario puede descifrarlo.

Los sistemas de clave pública también permiten garantizar la autenticación y el no repudio mediante la firma digital. Al cifrar un documento con nuestra clave privada, cualquiera puede descifrarlo con nuestra pública. Lógicamente, este proceso no tiene sentido para la confidencialidad pues cualquiera puede ver su contenido, pero sirve para demostrar la autoría de un documento, pues sólo el poseedor de la clave privada, puede haberlo firmado.



Los sistemas de cifra simétrica y asimétrica se complementan y tienen sus aplicaciones concretas de uso. Actualmente se utiliza en muchos escenarios la combinación de ambos, denominada **criptografía híbrida**, y que usamos a diario en nuestro navegador con el protocolo HTTPS cuando nos conectamos al banco, a redes sociales o a leer el correo electrónico vía web.Este sistema es la unión de las ventajas de los dos anteriores, debemos de partir que el problema de ambos sistemas criptográficos es que el simétrico es inseguro y el asimétrico es lento. El proceso para usar un sistema criptográfico híbrido es el siguiente (para enviar un archivo):

- Generar una clave pública y otra privada (en el receptor).
- Cifrar un archivo de forma síncrona.
- El receptor nos envía su clave pública.
- Ciframos la clave que hemos usado para encriptar el archivo con la clave pública del receptor.
- Enviamos el archivo cifrado (síncronamente) y la clave del archivo cifrada (asíncronamente y solo puede ver el receptor).

8.4. Seguridad activa

Por seguridad activa se entienden aquellas medidas que previenen e intentan evitar los daños en los sistemas informáticos. Son **medidas preventivas**. Para saber si una medida es seguridad activa o pasiva, hay que pensar si el incidente de seguridad ha ocurrido o no. Si se ha prevenido el incidente y no ha ocurrido gracias a las medidas de seguridad implantadas, se trata de seguridad activa. Si el incidente ya ha ocurrido y la medida minimiza el impacto del incidente o permite recuperar al sistema del fallo, se trata de seguridad pasiva.

Esta clasificación es comparable a las medidas de seguridad en los vehículos. Por ejemplo un ABS o ESP son ejemplos de medidas de seguridad activa pues intentan evitar el accidente, mientras que un arco de seguridad, un airbag o un cinturón de seguridad son medidas de seguridad pasiva pues intentan minimizar los daños una vez producido el accidente.

Como medidas o técnicas de seguridad activa en seguridad informática podemos citar los siguientes ejemplos:

- Contraseña: Acceso al sistema o aplicaciones por parte de personas no autorizadas.
- Permisos en ficheros: Acceso a ficheros por parte de personal no autorizado.
- Cifrado: Accesos no autorizados a información confidencial.
- Antivirus: Virus informáticos y en general, aplicaciones maliciosas
- Certificados digitales: Ataques a la integridad y autenticidad de los datos.
- Cuotas de disco: Fallo del sistema por agotamiento del espacio en disco.

Entre las medidas de seguridad activa que podemos aplicar, hay muchas como pueden ser:

- Proteger la BIOS o UEFI con contraseña
- Gestión de usuarios y política de contraseñas
- Autenticación de múltiples factores
- Gestión de permisos
- Cuotas de disco

8.5. Seguridad pasiva

Por seguridad pasiva, se entienden aquellas medidas que se aplican después de ocurrir un incidente de seguridad e intenta minimizar el impacto del mismo. Son **medidas correctoras o paliativas**. Por tanto el objetivo de esta medida no es evitar el incidente o la amenaza, que en algunos casos es imposible hacerlos, pero sí poder recuperar el sistema a su estado de funcionamiento anterior.

En la seguridad pasiva cabe destacar todas las medidas de redundancia y alta disponibilidad (HA, High Availability) sobre todo en centros de datos críticos, como pueden ser:

- Suministro eléctrico o conexión a Internet con más de un proveedor.
- Fuentes de alimentación dobles.
- Conmutadores y routers redundantes.
- Cableado y tarjetas de red redundantes.

Otros ejemplos de medidas de seguridad pasiva más habituales pueden ser:

- Discos redundantes: Restaura información de los otros discos en caso de fallo.
- Sistemas de ficheros tolerantes a fallos: Previene inconsistencia de datos en caso de apagones.
- SAI/UPS: Suministra energía al sistema en caso de un corte.
- Copias de seguridad: Recupera información en caso de pérdida o robo de información.
- Firma digital: Detecta un ataque a la integridad y autenticidad de los datos.
- Extintores, muros y paredes cortafuegos: Minimiza y palia los daños provocados por un incendio.

Entre las medidas de seguridad pasiva que podemos aplicar, podemos citar:

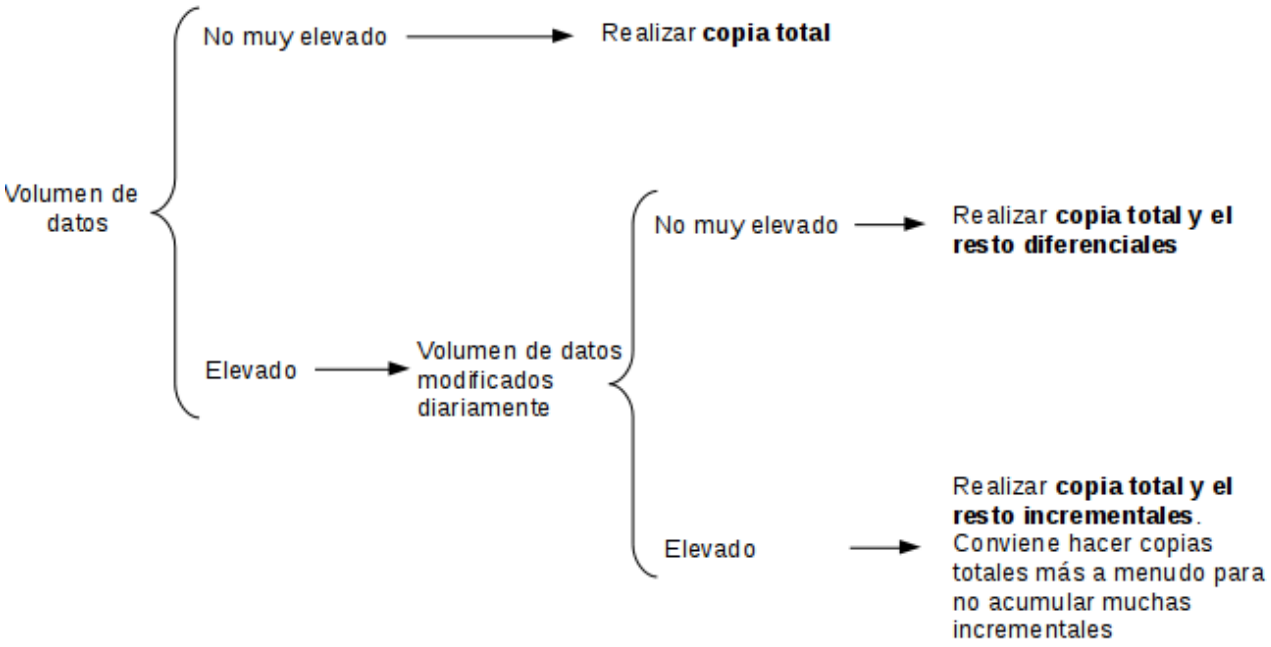
- Copias de seguridad
- Discos redundantes RAID
- Virtualización de sistemas operativos

Las copias de seguridad son una herramienta fundamental cuando hablamos de seguridad. Pueden salvar una empresa de un desastre como un incendio o un atentado en caso de destrucción de todos los datos. Las copias de seguridad forman parte de cualquier política de seguridad o plan de continuidad de negocio de cualquier organización.

En función de la cantidad de archivos que se salvaguardan a la hora de realizar la copia de seguridad, podemos distinguir tres tipos de copia:

- Copia de seguridad total o íntegra: Es una copia de seguridad completa de todos los archivos y directorios seleccionados
- Copia de seguridad incremental: Es una copia de seguridad sólo de los archivos que han cambiado desde la última copia de seguridad realizada. Si hacemos copia de seguridad total el día 1 de cada mes y copia de seguridad incremental el resto de los días, cada copia incremental solo guardará los archivos que se hayan modificado ese día. Si tenemos que realizar la restauración de archivos ante un desastre, debemos disponer de la copia total y de todas las copias incrementales que hayamos realizado desde la copia total.
- Copia de seguridad diferencial: Es una copia de todos los archivos que han cambiado desde la última copia de seguridad total que hayamos hecho. Si hacemos copia de seguridad total el día 1 de cada mes y copia de seguridad diferencial el resto de los días, cada copia diferencial guardará los archivos que se hayan modificado desde el día 1. La ventaja es que se requiere menos espacio que la copia total y que en el proceso de restauración únicamente necesitaremos la última copia total y la última copia diferencial. Por el contrario, se consume más tiempo en realizar la copia y también más espacio que en el caso de copia incremental.

El siguiente esquema resume los criterios a la hora de decantarse por algún tipo de copia,en función del volumen de datos a copiar:



A la hora de planificar las copias, en grandes organizaciones con un gran volumen de datos, se suelen usar sistemas mixtos. Por ejemplo en un caso típico se realizarían las siguientes tareas:

“ Todos los días 1 de cada mes, a las 23:00 horas: copia de seguridad total. Todos los viernes a las 23:00 horas: copia de seguridad diferencial desde la copia de día 1. Todos los días (excepto los viernes y el día 1) a las 23:00 horas: copia de seguridad incremental desde la copia del día anterior.”

Con ésta planificación nos aseguramos disponer de copia de seguridad diaria. En caso de desastre deberíamos recuperar la copia total, la última diferencial y todas las incrementales desde la última diferencial.

En las grandes organizaciones también es habitual utilizar caros sistemas hardware de backup centralizado basados en cabinas con cintas o soportes magnetoópticos, que suelen ir automatizadas con brazos robotizados que cambian los soportes para ir reemplazándolos en cada ciclo de copia.

Es importante destacar un problema que podemos tener al realizar las copias, que es la consistencia de los datos al estar copiando datos con los que se está trabajando. Es conveniente hacer las copias en horarios donde se sabe que no se está trabajando con los datos, por ejemplo en horarios fuera de oficina. En entornos más complejos, como data centers, donde se trabaja 24 x 7 y nunca se paran los servicios, se dispone de aplicaciones comerciales que suelen incluir agentes de copia para aplicaciones como los gestores de bases de datos Oracle o SQL Server, de forma que a la hora de lanzar la copia, realizan una instantánea de las tablas y registros de la base de datos de forma que se garantiza que los datos son coherentes en el momento de realizar la copia, aunque se esté trabajando con la base de datos.

Un **RAID** (Redundant Array of Independent Disks, antes Inexpensive Disks) es un sistema de almacenamiento que usa múltiples discos duros o SSD entre los que se distribuyen o replican los datos para proporcionar tolerancia a fallos. Tiene varias implementaciones y en función del uso es más conveniente una u otra.

Se pueden realizar por hardware o por software, siendo por hardware la opción más recomendada por rendimiento. Algunas controladoras RAID de servidores permiten cambiar los discos dañados en caliente sin apagar el servidor.

En los sistemas RAID de servidores es común hablar del disco hot spare, que es un disco de reserva que automáticamente forma parte activa del RAID si falla un disco, sin necesidad de intervenir el administrador.

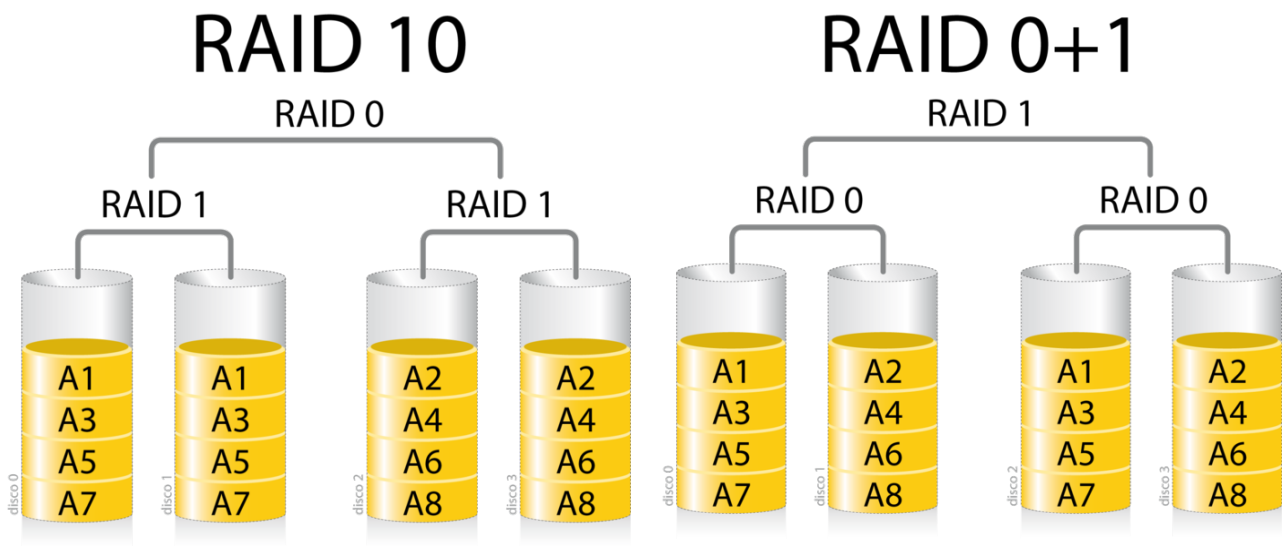
Algunos de los tipos RAID más usados.

- RAID 0:
  - Volumen de datos lógico a partir de discos físicos (se conoce también como stripping)
  - No hay redundancia de datos ni tolerancia a fallos: si falla un disco, perdemos datos (a no ser que tengamos copia, obviamente)
  - Mejora el rendimiento sobretodo en archivos grandes (p.ej: edición de vídeo)
  - Cada bloque se va escribiendo en un disco alternativamente
  - Se usan 2 o más discos
- RAID 1:

- Consiste en una copia exacta de un disco físico a otro (también se conoce como espejo o mirroring)
- Hay tolerancia a fallos y se pierde el tamaño de un disco (50%)
- Mejora el rendimiento de lectura
- Se forma con 2 discos

- RAID 5:
  - Divide los datos en bloques y usa paridad distribuida
  - La paridad permite recuperar los datos si se pierde un disco
  - Hay tolerancia a fallos y se pierde el tamaño de un disco (33% con 3 discos)
  - Mal rendimiento con escrituras pequeñas
  - Necesita mínimo 3 discos

Además existen los **RAID anidados**, que combinan varios tipos de RAID como el **10** o el **0+1**. En ambos casos se necesita 4 discos mínimo y deben ser pares. En el caso del 10 primero se hace el mirroring de datos cada dos discos, y luego el stripping sobre los dos RAID1. En el 0+1 es a revés: primero stripping y luego mirroring:



9. NORMATIVA LEGAL

Todas las empresas españolas y aquellas que operen en España están obligadas a cumplir una serie de normativas legales. Estas leyes están creadas para garantizar y proteger los derechos fundamentales de los ciudadanos. Este conjunto de normas están relacionadas con la gestión y protección de la información de sus usuarios y clientes, así como los sistemas informáticos que la tratan. Pero a pesar de que pueda existir una obligatoriedad en cumplir con la legislación, debemos de tener siempre presente que las mismas se desarrollan con el objetivo de ayudar a ciudadanos y empresas a hacer las cosas bien, y proteger el interés general mostrando las buenas prácticas para garantizar y proteger los derechos.

Dentro de las leyes principales que pudieran afectarnos desde el punto de vista de gestión y protección de la información son:

- **Ley Orgánica de Protección de Datos**, o **LOPD**. Vela por la seguridad de los datos personales que como empresa gestiona, ya sea en formato electrónico o papel. Esta ley aplica a la práctica totalidad de las empresas y autónomos, ya que todos ellos utilizan como mínimo datos de contacto del personal propio, clientes, o proveedores. A grandes rasgos, la LOPD nos obliga a notificar a la Agencia Española de Protección de Datos el tipo de datos que tratamos. Además también debemos de implantar diferentes medidas de seguridad según la sensibilidad de nuestra información.

- **El Reglamento General de Protección de Datos,** o **RGPD.** Es el reglamento europeo relativo a la protección de las personas físicas en lo que respecta al tratamiento de sus datos personales y a la libre circulación de estos datos

- **Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico**, o **LSSI**. Con esta ley se ofrecen garantías de seguridad en el comercio electrónico y transacciones online. Afecta principalmente a aquellas empresas con páginas web dedicadas a actividades lucrativas o económicas, a las que permitan la contratación online de servicios, ofrezcan información de productos, o se dediquen

al comercio electrónico. Sus requisitos de cumplimiento se limitan principalmente a requerir que incluyamos en nuestra página web, contratos y comunicaciones electrónicas diferente información.

- **Ley de Propiedad Intelectual**, o **LPI**. Con esta ley se crea un marco de protección legal para las obras de propiedad intelectual. Para ello define el concepto de «obra intelectual», crea un registro de obras y regula en qué términos se pueden o no utilizar estas obras según el criterio del autor.

Además debemos tener en cuenta que en función de nuestro sector de negocio y clientes, debemos cumplir otros aspectos regulatorios adicionales que podrían afectarnos como los siguientes:

- Ley 32/2003, General de Telecomunicaciones, en el caso de que tengamos relación con la prestación de servicios de comunicaciones electrónicas (correo electrónico).
- Ley 59/2003, de firma electrónica.
- Ley 17/2001, de Propiedad Industrial, si nuestra actividad implica la gestión de patentes y marcas.
- Ley 11/2007, de acceso electrónico de los ciudadanos a los Servicios Públicos, así como el Esquema Nacional de Seguridad (R.D. 3/2010, de 8 de enero) y el Esquema Nacional de Interoperabilidad (R.D. 4/2010, de 8 de enero) si trabajamos habitualmente con la Administración Pública.
- Ley 13/2011, de regulación del juego, si nuestra actividad tiene relación con el sector del juego online.
- Código de conducta para la prestación de los servicios de tarificación adicional basados en el envío de mensajes, en el caso de que prestemos servicios de almacenamiento y reenvío de mensajes sujetos a tarificación adicional.

También es importante comprender los **tipos de licencia** que ofrece el software. Estas licencias básicamente son un contrato entre el autor del programa y el usuario, y comprenden una serie de términos y cláusulas que el usuario deberá cumplir para usar el mismo. Algunos tipos de licencias de software son:

- **Software Libre o Free Software:** es un software disponible para cualquiera que desee utilizarlo, copiarlo y distribuirlo, ya sea en su forma original o con modificaciones. La posibilidad de modificaciones implica que el código fuente está disponible. Si un programa es libre, puede ser potencialmente incluido en un sistema operativo también libre. Es importante no confundir software libre con software gratis, porque la libertad asociada al software libre de copiar, modificar y redistribuir, no significa gratuidad. Existen programas gratuitos que no pueden ser modificados ni redistribuidos. Y existen programas libres de pago.

- **Copyleft:** la mayoría de las licencias usadas en la publicación de software libre permite que los programas sean modificados y redistribuidos. Estas prácticas están generalmente prohibidas por la legislación internacional de copyright, que intenta impedir que alteraciones y copias sean efectuadas sin la autorización del o los autores. Las licencias que acompañan al software libre hacen uso de la legislación de copyright para impedir la utilización no autorizada, pero estas licencias definen clara y explícitamente las condiciones bajo las cuales pueden realizarse copias, modificaciones y redistribuciones, con el fin de garantizar las libertades de modificar y redistribuir el software registrado. A esta versión de copyright, se le da el nombre de copyleft.

- **GPL:** la **Licencia Pública General GNU** (GNU General Public License GPL) es la licencia que acompaña los paquetes distribuidos por el Proyecto GNU, más una gran variedad de software que incluye el núcleo del sistema operativo Linux. La formulación de GPL es tal que en vez de limitar la distribución del software que protege, llega hasta impedir que este software sea integrado en software propietario. La GPL se basa en la legislación internacional de copyright, lo que debe garantizar cobertura legal para el software licenciado con GPL.

- **Freeware:** el término freeware no posee una definición ampliamente aceptada, pero es utilizada para programas que permiten la redistribución gratuita pero no la modificación. Estos programas no son software libre. En este tipo de licencia el autor puede restringir su programa al uso empresarial, redistribución no autorizada, modificación por usuarios y otro tipo de restricciones. (Ejemplos: Internet Explorer, Adobe Flash Player, Windows Live Messenger)

- **Shareware:** son programas distribuidos gratuitamente, pero por tiempo limitado o con algunos recursos restringidos. A través del pago de un valor definido por el autor del programa, se puede obtener el registro del programa o la versión integral con todos los recursos. Abarca las licencias adware, trial y demo.

- **Adware:** subprograma que descarga publicidad sobre otro programa principal. Esto ocurre cuando un programa tiene versiones comerciales o más avanzadas que necesitan ser compradas para poder ser utilizadas. Pagando por la versión comercial, esos anuncios desaparecen.

- **Trial:** versión de programa pago, distribuido gratuitamente con todos los recursos activos, pero por un tiempo determinado. Es como un programa freeware, que después de determinado tiempo deja de funcionar. Para continuar con la utilización del programa, se debe comprar la clave de registro e insertarla en el programa, para que vuelva a ejecutarse.

- **Demo:** versión de demostración, liberada por el autor, que no contiene todas las funciones del programa original. Es distribuida gratuitamente, y no tiene plazo de validez, pero también tiene pocos recursos y funciones.

- **Software Propietario:** el software propietario es aquel cuya copia, redistribución o modificación están, en alguna medida, prohibidos por su propietario. Para usar, copiar o redistribuir, se debe solicitar permiso al propietario o pagar.

- **Donationware:** versión de programa en la que el autor solicita una donación, para cubrir los gastos del desarrollo del programa. No es obligatoria, pero si solicitada. El programa no sufre variantes por realizar o no la donación.

- **Abandonware:** programa cuyo desarrollo fue abandonado. El autor debe anunciar públicamente el abandono del programa para ser abandoware, mientras tanto el programa está protegido contra los derechos de copia (copyright). Si la discontinuidad es anunciada, el programa puede ser distribuido y modificado por cualquier usuario o desarrollador.