FERNANDO J. MARTINI

Fernando J. Martini

Tecnología de la Información - Volumen 1 - El Hardware

TECNOLOGIA DE LA INFORMACION

VOLUMEN 1

EL HARDWARE

LA TECNOLOGÍA DE LA INFORMACIÓN

VOLUMEN 1

EL HARDWARE

Versión: 1.03

Martini, Fernando J.
Tecnología de la Información: el hardware . - 1a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2014.
v.1, CD-ROM.
ISBN 978-987-33-4692-7
1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

Martini, Fernando J.
Tecnología de la Información. - 1a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2014.
v.3, CD-ROM.
ISBN 978-987-33-4691-0
1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

Martini, Fernando J.
Tecnología de la Información. - 2a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2018.
v.3, CD-ROM.
ISBN 978-987-33-4691-0
1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

ÍNDICE

Capítulo III: Esquema y funcionamiento básico de una computadora

1-Funcionamiento en general
1.1-Tablas de conversión de codificación humana a codificación binaria
1.2-El cálculo matemático en circuitos electrónicos
1.3-Algebra proposicional o compuertas lógicas
1.4-Circuito sumador con simbología proposicional
1.5-Una compuerta lógica AND implementada con transistores NMOS y PMOS
2-Diagrama básico de unidades funcionales de la computadora
2.1-La CPU
2.2-La memoria primaria o RAM
2.3-El bus
2.4-Buses y periféricos

Capítulo IV: Discos rígidos, cintas magnéticas, discos ópticos y tarjetas de memoria

1-Discos duros
2-Unidades de estado sólido
3-Comparación entre HDD y SDD
4-Cinta magnética
5-Discos ópticos
5.1-El CD
5.1.1-El CD-ROM
5.1.2-El CD-R
5.1.3-CD-RW
5.2-DVD
5.2.1-DVD-ROM
5.2.2-DVD-R
5.2.3-DVD+R
5.2.4-DVD-RW
5.2.5-DVD+RW
5.3-BLU RAY
6-Tarjeta de memoria o memoria flash
6.1-Las tarjetas microSD

Capítulo V: Códigos de barras, códigos de barra bidimensionales, código QR, MICR, escáner, reconocimiento de voz, RFID, cámaras digitales, teclado, mouse, touchpad, sensor de huellas digitales.

Capítulo I
Conceptos elementales

1-Las primeras máquinas de calcular mecánicas

Indudablemente, realizar cálculos con el contador o la regla de cálculo no resultaba muy satisfactorio para los requerimientos de las ciencias hacia fines del siglo XIX y principios del siglo XX. La primera máquina que logró satisfacer en gran medida los fuertes requisitos de esa época fueron las máquinas de ruedas dentadas que ayudaron en mucho a la realización de las operaciones matemáticas básicas. En 1872, Frank Baldwin inventó en los Estados Unidos la calculadora de rueda dentada, que también fue desarrollada independientemente dos años después por W.T. Odhner, en Suecia. El modelo de Odhner y otros similares de otras compañías fueron vendidos en varios miles de unidades en los años 70.

Una máquina avanzada y muy vendida de estas características fue la famosa calculadora Facit (Ver Figura 1) que permitía sumar, restar, multiplicar y dividir con mucha sencillez.

Figura 1

El funcionamiento de estas máquinas era sumamente simple, básicamente consistía en ruedas de diez dientes que se colocaban en un mismo eje (en el caso de la Facit, tenía trece ruedas sobre un eje para el número resultante de las operaciones de suma o resta). La rueda de las unidades (el primer dígito a la derecha), luego de girar diez dientes, hacía girar la rueda de las decenas un diente; cuando la rueda de las

decenas rodaba diez dientes, hacía girar un diete la rueda de las centenas, y así sucesivamente.

Por ejemplo, de este mecanismo de ruedas dentadas, se puede observare el interior de una máquina de sumar muy antigua, pero muy eficiente y simple, como es el caso de la Golden Gem (Ver Figura 2).

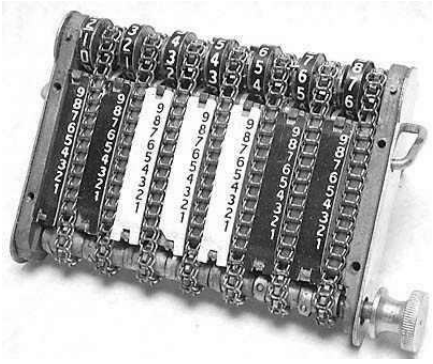


Figura 2

Hasta aquí la tecnología requerida no iba mucho más lejos de la capacidad del hombre para fundir metales, crear aleaciones, moldearlas a su necesidad y con una precisión milimétrica; aunque a la humanidad le haya costado más de tres millones de años llegar hasta este punto.

Pero, para llegar a las primeras computadoras electrónicas comerciales (por ejemplo, la UNIVAC-1 (1951)), se ha tenido que unir una cantidad importante de conocimiento que a lo largo de más de dos mil años se ha estado acumulando. El principal de todos los conocimientos adquiridos que permitieron el desarrollo de los actuales procesadores ha sido el átomo.

2-El átomo

Algunos cientos de años antes de Cristo, en la antigua Grecia, los filósofos debatían si un trozo de una piedra cualquiera, podía dividirse infinitamente y ser siempre una parte más pequeña de la piedra original. Aproximadamente en el 450 a. de C., el filósofo Leucipio sostuvo que un material podía ser dividido, pero hasta un momento en que la partícula ya sería tan pequeña que no podría seguir dividiéndose. Un discípulo suyo, Demócrito, llamó a esta partícula átomo, que significaba

«indivisible». Esta doctrina fue denominada Atomismo. Estos filósofos no podían comprobar estas teorías, que eran puramente deducciones argumentales.

Lamentablemente, para muchos otros filósofos de esa época, incluido Aristóteles, la concepción de una partícula de materia no divisible en otras menores resultaba paradójica y no fue compartida. Por lo que, si bien el Atomismo no murió del todo, tuvo muy pocos adeptos hasta dos mil años después.

Hoy sabemos que un átomo de carbono mide aproximadamente 340 picómetros (0.000000000340 metros). Recientemente, gracias a Steven Chu, Premio Nobel de Física (1997), ahora el límite para ver cosas en un microscopio óptico está en torno a los 200 nanómetros. Teniendo en cuenta que 1 nm = 1000 pm, aún estamos lejos de poder ver con nitidez un átomo. Pero a pesar de este inconveniente, gracias a experimentaciones y deducción lógicas, hoy sabemos mucho de los átomos y este conocimientos es el que nos ha permitido el impresionante desarrollo de la tecnología en general y, particularmente, de la tecnología de la información.

Para que se tenga una idea más acabada del tamaño del átomo, se lo comparará con un grano de café:

- Grano de café: 12 x 8 mm
- Grano de sal: 0,5 mm
- Una ameba: 500 µm (micrómetros)
- Un paramecio: 210 x 60 µm
- Una célula de la piel: 30 µm
- Un glóbulo rojo: 8 µm
- Una bacteria Ecoli: 3 x 0,6 µm
- El virus de la influenza: 130 nm (nanómetros)
- Un anticuerpo: 12 nm
- Una molécula de agua: 275 pm (picómetros)
- Un átomo de carbón 340 pm

Se agrega una escala métrica para facilitar el entendimiento:

Valor	Símbolo	Nombre
10^{-1} m	**dm**	**decímetro**
10^{-2} m	**cm**	**centímetro**
10^{-3} m	**mm**	**milímetro**
10^{-6} m	**µm**	**micrómetro**
10^{-9} m	**nm**	**nanómetro**
10^{-12} m	**pm**	**picómetro**

3-La estructura del átomo

El átomo se compone, fundamentalmente, de dos partes: núcleo y corteza. En el núcleo, se encuentran los **protones**, los que están cargados positivamente, y los n**eutrones,** que como su denominación lo indica son neutros. En la corteza se encuentran los **electrones,** que están cargados negativamente y orbitan alrededor de los protones (Ver Figura 3. Átomo de oxígeno). Ya se han efectuados desarrollos científicos que demuestran que dentro del átomo hay otras partículas más, como los quarks, pero no nos detendremos en ellos.

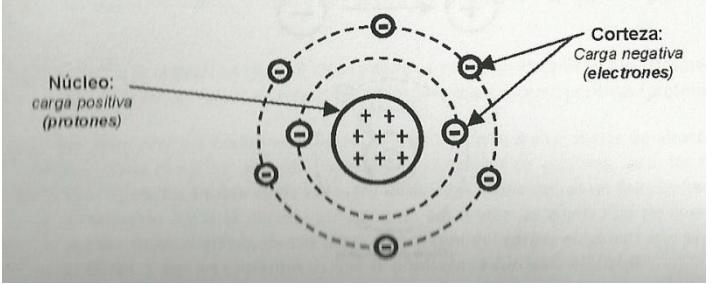


Figura 3

Todos los átomos en estado de equilibrio tienen igual número de protones que de electrones, por lo que se dice que el átomo, en este estado, es eléctricamente neutro. Tanto los protones (+) como los electrones (-) tienen cierta energía que ejerce una fuerza que hace que las cargas del mismo signo se repelan y las cargas de distinto signo se atraigan. Es decir, un protón con otro protón, así como un electrón con otro electrón, tienden a repelerse. Mientras que un protón con un electrón tienden a atraerse. Este efecto es el que mantiene unidos a los protones con los electrones.

Los símbolos + y – se utilizan para indicar los dos tipos de estados eléctricos (o polaridades).

4-La corriente eléctrica

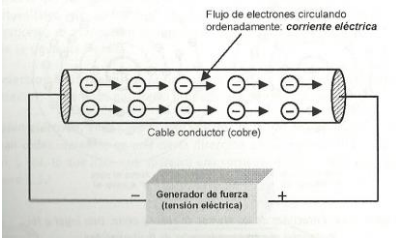
Los electrones son atraídos por los protones que no llegan a este y continúan su trayectoria orbital debido a otras fuerzas que originan el movimiento rotativo a gran velocidad. También, es importante entender que los electrones circulan en distintas capas orbitales. Algunas capas están más cerca del núcleo y otras más alejadas. Los electrones ubicados en las capas orbitales más alejadas del núcleo están atraídos más débilmente por los protones del núcleo que los electrones de las capas más cercanos a este.

Algún tipo de energía externa puede hacer que los electrones más separados del núcleo, los de la última capa, abandonen el átomo, dando lugar al concepto de cargas eléctricas móviles, y son la causa de que se produzca la corriente eléctrica.

El electrón que se escapa de un átomo y pasa a otro átomo produce que el primero quede con carga positiva (el que perdió el electrón) y el átomo que recibe el electrón quede con carga negativa. Esto explica que pueda haber cuerpos cargados positivamente y otros cargados negativamente.

La corriente eléctrica se produce por el movimiento de electrones y se puede definir como la circulación ordenada de electrones a través de un conducto (Ver Figura 4).

Figura 4



5-Conductores y aislantes

Existen materiales cuyos átomos tienen en su última capa un solo electrón, por lo que este es muy propenso a que por poca fuerza abandone el átomo. Por ejemplo, es lo que ocurre con el átomo de cobre y el átomo de plata (Ver Figura 5).

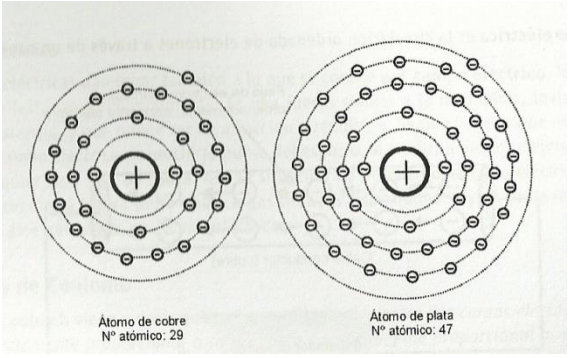


Figura 5

Los materiales aislantes, por el contrario, tienen átomos con muchos electrones en su última capa, por lo que es casi imposible que se separen del núcleo. Por ejemplo, la porcelana, la goma, etcétera.

6-Fuerza electromotriz

Los generadores de electricidad (como las pilas, las baterías, los alternadores, etcétera) fuerzan a que los electrones circulen ordenadamente por un circuito establecido. Esta fuerza electromotriz se mide en voltios. La velocidad de la transmisión de la electricidad es de aproximadamente 300.000 km/s. Esta velocidad es alcanzada porque todos los electrones libres se pasan de átomo en átomo a la vez (Ver Figura 6).

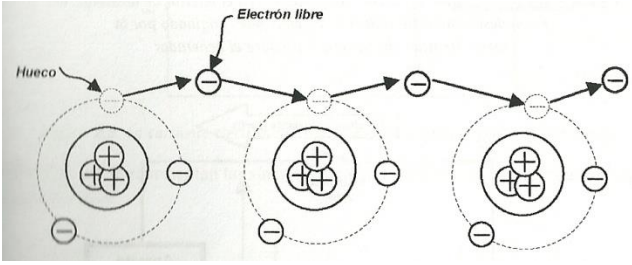


Figura 6

Como se ve en esta última gráfica, los electrones (negativos) empujados por la fuerza electromotriz se pasan hacia el átomo contiguo y así sucesivamente.

En la figura 7, se puede ver un circuito con un cable conductor en el que se produce un flujo de electrones desde el negativo hacia el positivo. Cuando salta un electrón de un átomo, deja algo así como un hueco positivo. Esto es un átomo cargado positivamente. Este hueco será ocupado por otro electrón, lógicamente negativo, hasta que vuelva a ser empujado hacia el próximo átomo. Las líneas perpendiculares del circuito representan el elemento que está causando la fuerza electromotriz.

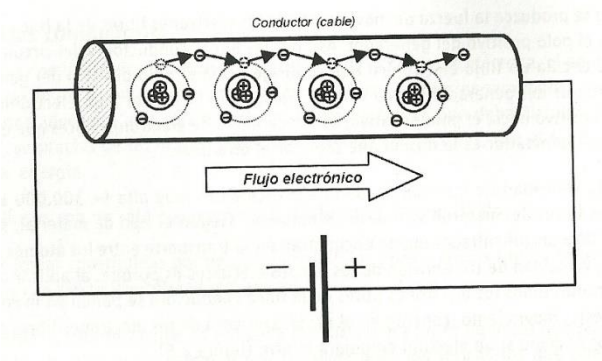


Figura 7

7-Una simple forma de generar fuerza electromotriz

La forma más simple para generar fuerza electromotriz es con dos elementos: un alambre de cobre y dos imanes (Ver Figura 8).

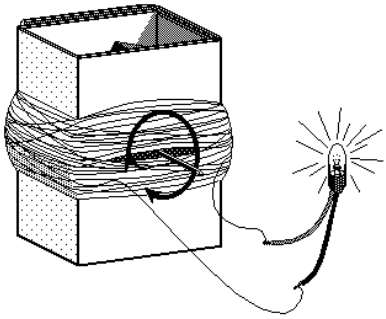


Figura 8

Se arma una simple caja de cartón a la que se envuelve con alambre de cobre. Los extremos del alambre de cobre se unen a los dos polos de una simple lamparita de linterna. La caja es atravesada por un clavo que dentro de la caja de cartón sostiene dos imanes. Al girar el clavo junto con los imanes, se produce un campo de fuerza (campo magnético) que provoca que los electrones de la última capa del alambre de cobre empiecen a saltar de uno a otro. Así, se produce lo que se denomina corriente eléctrica y esta corriente prende la bombita unida a los dos polos. A este aparato se lo denomina dínamo. Varios de estos experimentos pueden verse en YouTube.

El magnetismo es un estado caracterizado por un campo de fuerza que puede actuar sobre algunos materiales, es invisible y de acción a distancia. En la naturaleza, existe un mineral de hierro llamado magnetita que tiene propiedades magnéticas de forma natural. Algunos materiales ferrosos son susceptibles de ser imantados si sobre ellos se ejerce una fuerza magnética externa. Cada trozo de imán tiene un polo norte y un polo sur. Los polos iguales ejercen una fuerza de repulsión y los polos opuestos una de atracción.

En la figura 9, se puede observar otro ejemplo muy sencillo de una dínamo.

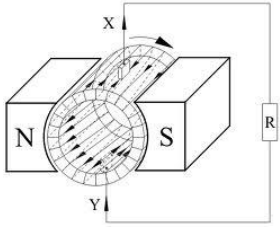


Figura 9

En este ejemplo, el campo de fuerza de los imanes es ejercido sobre una bobina con alambre de cobre que gira, e igual que en el ejemplo anterior, esta fuerza provoca que se produzca la corriente eléctrica.

El cobre fue uno de los primeros metales utilizados por el hombre en la prehistoria. La magnetita fue descubierta aproximadamente 2500 años a. de C. Pero, pese a que los elementos se conocían desde mucho tiempo atrás, recién en el año 1831 el físico químico británico Michael Faraday fue quien desarrolló el primer generador eléctrico.

8-Corriente continua y corriente alterna

Según la forma como se genere el flujo de corriente, esta puede ser continua o alterna. La corriente eléctrica generada por generadores eléctricos es alterna. Las pilas, baterías y otros generan corriente eléctrica continua.

En la corriente continua, los electrones siempre circulan en el mismo sentido de dirección; mientras que en la corriente alterna, los electrones no se desplazan de un polo a otro, sino que a partir de su posición fija en el cable (centro), oscilan de un lado al otro de su centro, dentro de un mismo entorno o amplitud, a una frecuencia determinada (número de oscilaciones por segundo). En la Argentina, la red doméstica de 220 V alterna oscila a 50 ciclos por segundo (50 Hz).

9-Electromagnetismo

El flujo de electrones produce lo que se denomina un campo magnético (o fuerza magnética). La primera experiencia que puso de manifiesto que el movimiento de cargas eléctricas (la electricidad) da origen a fuerzas magnéticas fue realizada en 1819 por el físico danés Hans Christian Oersted. Su experimento consistió en poner una brújula cerca de un cable energizado por una pila voltaica (descubrimiento anterior al generador eléctrico y que se desarrolló mediante una reacción química entre la unión del cobre y el zinc) y observar que la aguja se movía (Ver Figura 10).

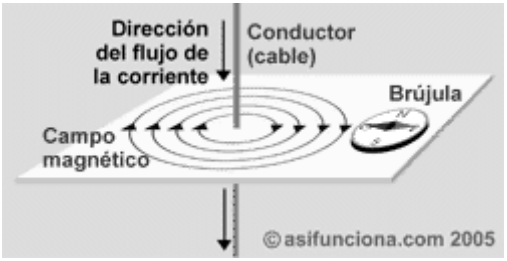


Figura 10

10-Un sencillo motor eléctrico

En la figura 11, se puede ver un anillo o rollo de alambre de cobre conectado de un lado al polo positivo de la pila y del otro lado, al borne negativo. Este anillo de cobre se encuentra suspendido a unos centímetros de un imán por unos alambres conductores que trasmiten la corriente y lo sostienen para poder girar. Cuando la corriente fluye por el rollo de cobre, se provoca una fuerza magnética a su rededor que al contrarrestarse con la fuerza del imán lo obliga a girar sobre su eje. Muchos de estos ejemplos se pueden observar en YouTube.

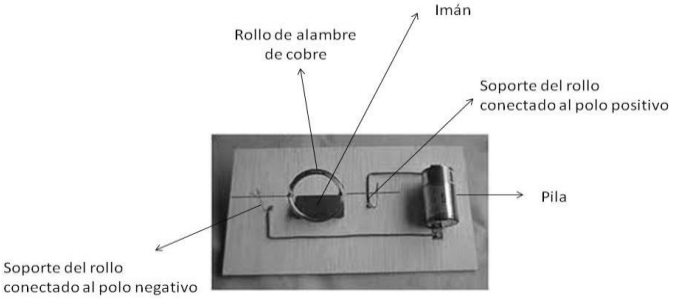


Figura 11

Capítulo II
Componentes básicos

1-La bobina

Es un hilo conductor (normalmente de cobre) enrollado en algún tipo de soporte (generalmente un tubo de material ferromagnético), también puede estar enrollado en el aire. Al circular la energía por la espiral, esta se comporta como un imán con un polo magnético en cada extremo de la espiral. A estos también se los denomina electroimanes (Ver Figura 12).

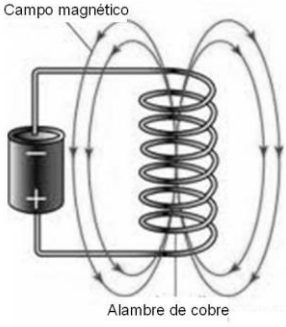


Figura 12

2-El electroimán como contactor

En la figura 13, se esquematiza un electroimán que funciona como contactor. Al pasar corriente por la bobina, el material ferroconductor se magnetiza y atrae la armadura que cierra el contacto. Cuando se suspende la energía, la armadura debe retraerse.

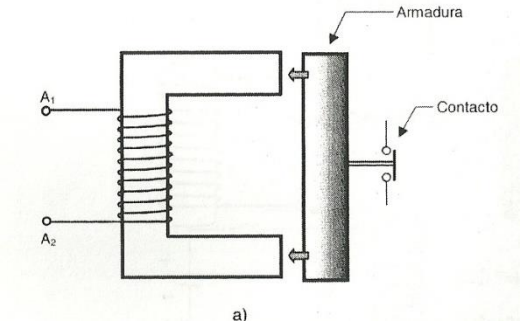


Figura 13

3-El transformador

Como se pudo ver, la circulación de electrones produce fuerza magnética y, también, se observó que una fuerza magnética aplicada sobre un conductor de cobre genera electricidad.

El transformador es un instrumento basado en el electromagnetismo, que permite variar en forma muy sencilla los valores de tensión (Ver Figura 14).

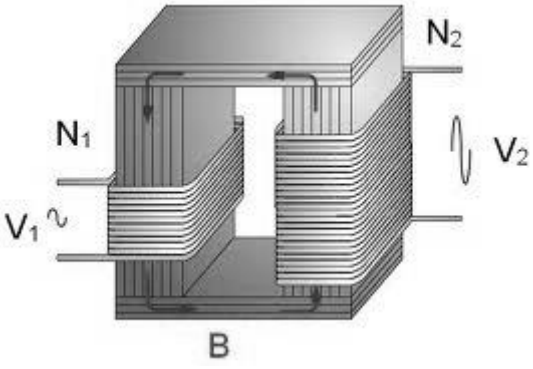


Figura 14

El sistema se basa en dos bobinas unidas por un núcleo ferromagnético. Al ingresar una tensión de entrada en la bobina primaria o V1, se genera un flujo magnético a

través del cuerpo ferromagnético que se acopla con la bobina secundaria V2 y, entonces, se produce una corriente de salida que, en función del bobinado secundario, será menor, mayor o igual que la corriente de entrada.

4-Resistencias

Todos los materiales y elementos conocidos ofrecen mayor o menor resistencia al paso de la corriente eléctrica, incluidos los mejores conductores.

La resistencia de un conductor está determinada por las propiedades de la sustancia específica (cobre, plata, bronce, etcétera), por la longitud del conductor, por la superficie transversal del objeto, así como por la temperatura.

La resistencia eléctrica tiene múltiples aplicaciones, por ejemplo: en una estufa, producen calor, el filamento de una lámpara produce luz, etcétera. Es importante dejar claro que estas resistencias no consumen electrones. Lo que consumen es la energía necesaria para hacer circular los electrones. Cuando pasan los electrones por un filamento tan delgado como el tungsteno en una bombilla eléctrica, lo que ocurre es que, al rozar los electrones con el filamento, se produce calor, tanto que emiten una intensa luz. La energía consumida es el esfuerzo de hacer pasar los electrones por ese filamento.

Imagínese a la resistencia como una llave que se opone al paso de la corriente de agua, en un circuito sería el equivalente del potencial eléctrico que también se llama voltaje. Si se tiene una batería de 9 volts, y se desean solamente 3, entonces es necesario colocar una resistencia que baje la potencia de 9 a 3 volts. La resistencia en este caso no permitirá el paso de los electrones equivalentes a 6 volts.

Se las gráfica como se muestra en la figura 15 y se fabrican de múltiples materiales.

Figura 15

5-El condensador

Un condensador, también llamado capacitor, es un dispositivo utilizado para almacenar energía y es muy utilizado como memoria de computadoras. Consiste en dos armaduras metálicas (láminas conductoras) separadas por un material aislante llamado dieléctrico (Ver Figuras 16 y 17).

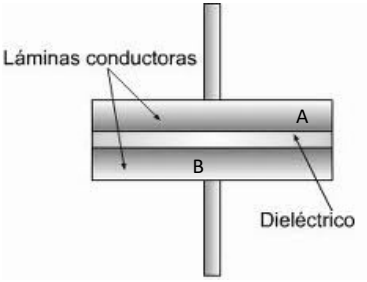


Figura 16

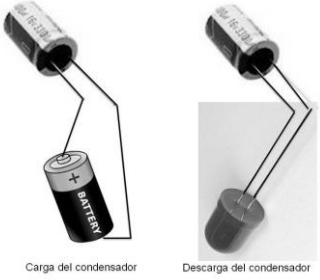


Figura 17

En un principio, las armaduras (o láminas conductoras) del condensador contienen idéntico número de cargas positivas y negativas. Todos sus átomos están en equilibrio eléctrico.

Si conectamos la lámina A al extremo de una pila y la lamina B al otro extremo de la pila, una lamina quedará cargada positivamente y la otra negativamente. La diferencia de potencial entre las láminas es la carga eléctrica que obtiene el condensador. Por más que desconectemos al condensador de la pila, este continuará

por un lapso de tiempo cargado de idéntica forma (salvo que esté totalmente aislado y entonces sí mantendrá la carga por mucho tiempo).

Si, por ejemplo, se une con un cable la lámina A y la lámina B a un led, ocurrirá que las dos láminas buscarán el equilibrio eléctrico, por lo que por el cable circulará una corriente eléctrica que prenderá el led hasta que el equilibrio se obtenga, y entonces el condensador quedará descargado.

6-La electrónica

Hay muchas definiciones de electrónica, todas varían sustancialmente, para esta obra se tomará la siguiente: «La electrónica es la rama de la física, y la especialización de la ingeniería, que estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el control del flujo microscópico de los electrones u otras partículas cargadas eléctricamente».

Todo lo ya descripto en este libro tiene que ver con la electrónica, pero también con la electricidad en general. Se desarrollarán a continuación algunos componentes fundamentalmente vinculados a la electrónica y de conocimiento imprescindible para entender el hardware de las computadoras.

7-Semiconductores

Los semiconductores están realizados con materiales que se encuentran entre una situación intermedia entres los buenos conductores de energía eléctrica y los que son aislantes. Con la particularidad de que la conductividad aumenta con el aumento de la temperatura. Estos materiales se caracterizan porque tienen 4 electrones en la última capa del átomo, como el silicio y el germanio. Los semiconductores son la base para la construcción de los diodos y los transistores.

8-Enlace covalente

Para simplificar el análisis, se considerará únicamente la última capa de 4 electrones de los materiales semiconductores (Ver Figura 18).

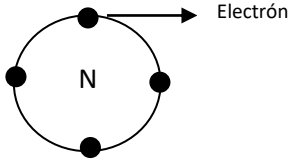


Figura 18

En una placa de silicio, los átomos están enlazados entre sí de forma que comparten los 4 electrones de su última capa (Ver Figura 19).

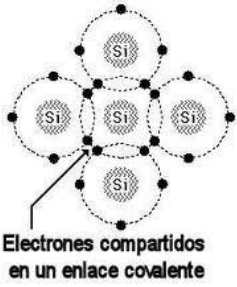


Figura 19

Como se puede ver, el átomo del medio comparte sus electrones de la última capa con los átomos lindantes. Si se ampliase la muestra, se verá que esto ocurre con el resto de los átomos. Si bien cada neutrón mantiene su pertenecía y orbita dentro de su átomo, este camino orbital lo comparte con los neutrones de los átomos lindantes. Este tipo de enlace se llama covalente.

9-Huecos

Si por algún motivo, como puede ser el calentamiento del material, algún electrón escapa del enlace covalente, se genera un hueco. Este hueco genera una carga positiva porque le falta el electrón negativo que mantenía el equilibrio eléctrico. Este hueco puede ser buscado por cualquier otro electrón suelto para instalarse en ese lugar (Ver Figura 20).

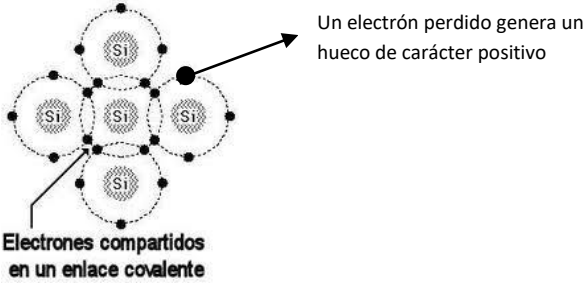


Figura 20

10-Los semiconductores N y P

Mediante una técnica denominada dopado, se obtienen semiconductores de diferente grado de conductividad. Este dopado consiste en introducir en el semiconductor diferentes estructuras atómicas; esta es la forma en que se construyen los semiconductores N y P.

Mediante la unión de semiconductores N y P, se fabrican los diodos y los transistores utilizados actualmente. Estos semiconductores son fundamentales para construir los actuales circuitos de las computadoras.

Hoy existen muchos materiales para construir semiconductores N y P, en esta obra se utilizará como ejemplo el silicio dopado con fósforo o boro.

11-Semiconductores N

El semiconductor N se obtiene introduciendo una cantidad adecuada de fósforo en el material de silicio. El fosforo es un material cuyos átomos tienen 5 electrones en su última capa. Por lo tanto, un átomo de fosforo puede enlazarse con el átomo de silicio utilizando 4 electrones, por lo que le quedará un electrón libre. Esto hace que el material logrado tenga una carga negativa (N) (Ver Figura 21).

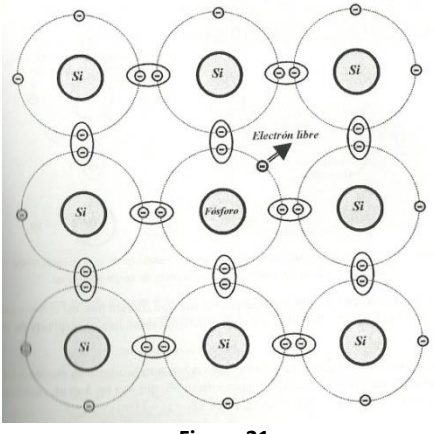


Figura 21

12-Semiconductor P

El semiconductor P se obtiene introduciendo una adecuada cantidad de boro al material de silicio. Como los átomos de boro están formados por tres electrones en

su última capa, al enlazarse con los átomos de silicio, quedan huecos, como se puede ver en la figura 22.

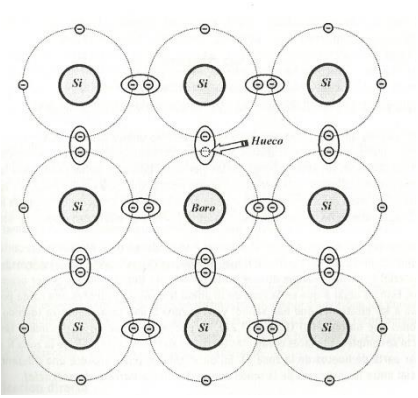


Figura 22

13-El diodo

Es un componente que forma parte de la mayoría de los circuitos electrónicos. Está construido con materiales semiconductores. El diodo se utiliza fundamentalmente para dejar correr la energía en un solo sentido, por eso es que se utiliza para convertir la corriente alterna en corriente continua. Dar una explicación con rigor científico implicaría una extensión y complejidad que no es la que se busca para este trabajo. Pero sí daremos un acercamiento al concepto del funcionamiento del diodo.

El diodo se fabrica mediante la unión de un semiconductor P con un semiconductor N (Ver Figura 23). Para tener una idea de qué tamaño posee un diodo, cabe aclarar que en un pequeño chip de 1 cm x 1 cm puede haber millones de diodos.

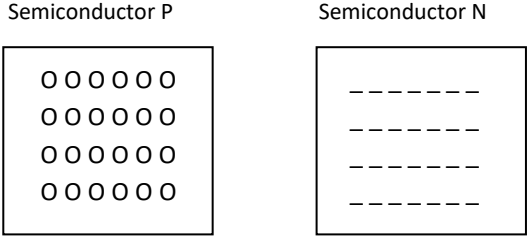


Figura 23

Cuando el cristal del semiconductor P se une con el cristal de semiconductor N, en las inmediaciones de la unión, se produce un traspaso de electrones del cristal N. Los electrones débiles del fósforo, que no están en las uniones covalentes, pasan a los huecos de los covalentes del cristal P (por el faltante que se produjo en la introducción del boro). Esto produce una diferencia de potencial inicial. No todos los electrones libres del semiconductor N logran alcanzar algún hueco en el semiconductor de tipo P. Llega un momento en que se produce **una barrera de potencial** donde la fuerza se equilibra (Ver Figura 24).

Para que se entienda el diodo, se dará un ejemplo con una diferencia de potencia cercana a los 0,65 V.

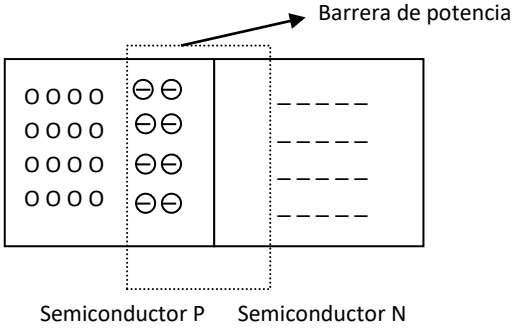


Figura 24

Ahora, qué pasa si se conecta una fuerza de corriente eléctrica de unos 0,7 V (un poco más alta que el diferencial de potencia de la barrera) para que intente ingresar electrones por la zona P y retirar electrones de la zona N (Ver Figura 25). Con esa fuerza, los electrones ingresantes logran moverse por los huecos de la zona P, luego sobrepasan la barrera de potencia y, por último, logran hacer mover los electrones sobrantes del fósforo en la zona N. De esta forma, se establece el sentido de la corriente.

En cambio, si se cambia la polaridad, es decir, que se intenta ingresar electrones por la zona N, estos moverán los electrones de la zona N y engrosarán la barrera de potencia, por lo cual el diferencial de potencial será cada vez mayor y la necesidad de corriente a introducir mayor también. Llegará un momento en que la barrera podrá vencerse, pero no sin lograr un gran deterioro del diodo o su destrucción total.

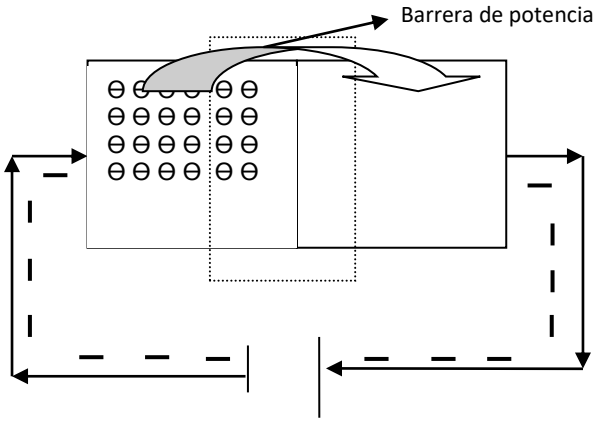


Figura 25

14-Los transistores bipolares para amplificación

El transistor bipolar, al igual que los diodos, está formado por semiconductores de tipo N y P. Es un dispositivo de tres terminales: emisor, colector y base. Atendiendo a su fabricación, puede ser de dos tipos: NPN y PNP. En este caso, se verá el transistor NPN por ser el más usado (Ver Figura 26).

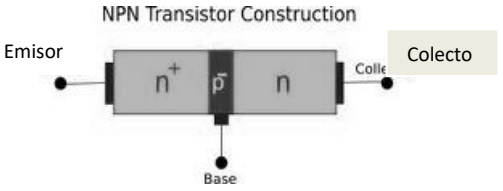


Figura 26

La región del emisor ha de ser muy dopada, es decir, con gran densidad de átomos de fósforo, lo que hace que haya muchos electrones sueltos. La base debe ser muy estrecha y poco dopada (pocos huecos). La región del colector debe estar menos dopada que la del emisor.

Este transistor puede verse como dos diodos en oposición. Como en el caso de los diodos, entre el sector P y el N+ existe una barrera de potencia, lo mismo que entre la zona P y N. (Ver Figura 27)

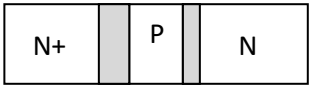


Figura 27

El circuito de amplificación se realiza de la siguiente forma (Ver Figura 28):

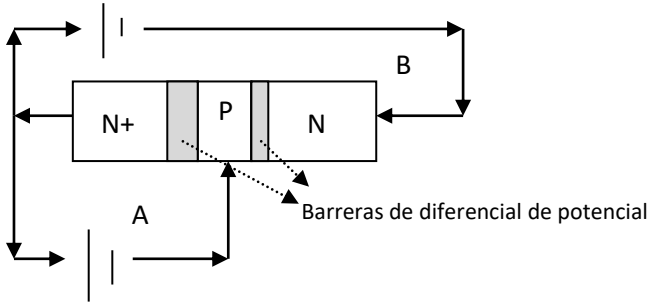


Figura 28

Una corriente muy baja ingresa por el circuito A a la zona P (como puede ser la señal de un micrófono). Estos electrones cubren los huecos de la zona y bajan el diferencial de potencial de las barreras. De esta forma, permiten que la corriente proveniente del circuito B, mucho más elevada que la del circuito A, ingrese por la zona N, se sume a la corriente ingresada en la zona P y supere la segunda barrera, generando la señal del circuito A amplificada.

15-Los transistores NMOS y PMOS

Se describirá brevemente este tipo de transistores, ya que son hoy los pilares de los procesadores y de las memorias de las computadoras. Se utilizan, básicamente, como compuertas (más adelante se verá esta funcionalidad), que es el fundamento de la computación electrónica.

En la figura 29, se puede ver un transistor de tipo NMOS que normalmente se utiliza como compuerta y del que suele haber millones en cada microprocesador. Se

denomina así por la unión metal-óxido-semiconductor. Está formado por una lámina semiconductora de tipo P. Además, tiene dos pequeñas islas o porciones de semiconductor tipo N (de la cual deriva el nombre), una bajo el contacto fuente y otra bajo el contacto sumidero o salida. Por debajo de la capa N, se encuentra un contacto metálico.

Al unirse los semiconductores tipo N con los tipo P, se producen unas regiones de diferencial de potencial idéntica a la que se producen en el caso del diodo. Además, se puede ver una capa de óxido, normalmente SiO2 (óxido de silicio). Esta capa es un material denominado dieléctrico, como se utiliza en los condensadores. Aquí se aplica aprovechando su capacidad aislante para que los conectores de fuente, puerta y sumidero no se contacten entre sí y le quiten la funcionalidad al transistor. Pero, también es un material que no requiere mucha fuerza eléctrica para traspasarla.

La función de este transistor es dejar circular o no la corriente entre la fuente (S) y el sumidero (D) o viceversa, ya que el transistor es simétrico. Para permitir que fluya la corriente, es necesario utilizar la puerta G. Para abrir la compuerta, es necesario enviarle una pequeña corriente. Una cantidad importante de electrones alcanzará la zona P que se coloca en los huecos de la zona de ella, entre las puertas S y D, formando el canal L. De esta forma, se elimina la diferencia de potencial de las barreras y se permite el paso de la corriente eléctrica entre las puertas S y D (Ver Figura 30).

Por lo tanto, cuando se envían neutrones por la puerta G (polaridad negativa), el transistor está abierto (fluye corriente); y cuando la puerta está en 0 V, el transistor está cerrado.

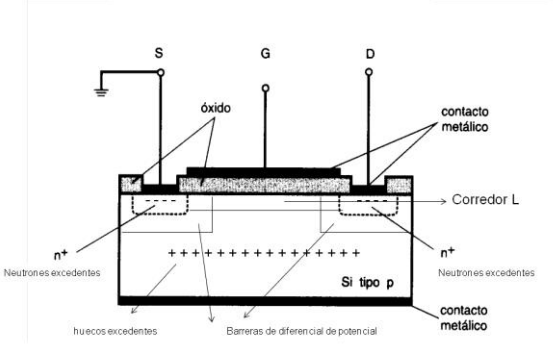


Figura 29

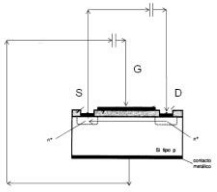


Figura 30

Los transistores PMOS tienen las mismas funcionalidades que el NMOS. La diferencia con el NMOS es que los semiconductores que conectan las puertas S y D están fuertemente dopadas positivamente (zonas P), y el cuerpo del transistor es un semiconductor tipo N (Ver Figura 31).

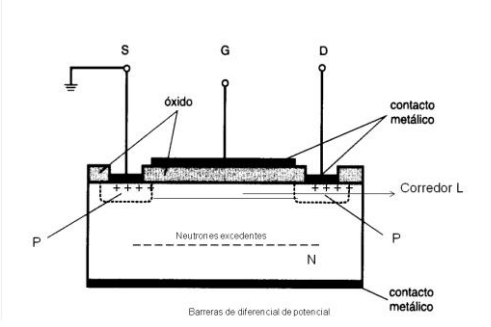


Figura 31

Al inverso que los transistores NMOS, los PMOS, para ser accionados, necesitan que su puerta G esté conectada en forma inversa que los NMOS, es decir, que la puerta G tiene que tener una carga positiva.

16-Led

Los led (*Light-Emitting Diodeson*) son diodos que cuando circula la corriente eléctrica emiten luz. Todos los diodos emiten luz, lo que ocurre es que en los diodos de silicio la emisión de luz es muy tenue. Pero existen otros materiales semiconductores con los cuales se pueden construir diodos que emiten luz y de diferentes características (Ver Figura 32). A continuación, se muestra una tabla de materiales y su emisión de luz.

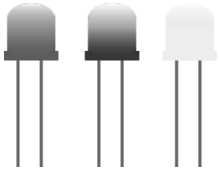


Figura 32

Compuesto	Color
Arseniuro de galio (GaAs)	Infrarrojo
Arseniuro de galio y aluminio (AlGaAs)	Rojo e infrarrojo
Arseniuro fosfuro de galio (GaAsP)	Rojo, anaranjado y amarillo
Fosfuro de galio (GaP)	Verde
Nitruro de galio (GaN)	Verde
Seleniuro de zinc (ZnSe)	Azul
Nitruro de galio e indio (InGaN)	Azul

Carburo de silicio (SiC)	Azul
Diamante (C)	Ultravioleta

17-El rayo láser

El láser es un dispositivo electrónico que amplifica un haz de luz a una extraordinaria intensidad. Se basa en la excitación de una onda estacionaria entre dos espejos, uno opaco y otro traslúcido, en un medio homogéneo. Como resultado de este proceso, se origina una onda luminosa de múltiples idas y venidas entre los espejos, que sale por el traslúcido.

El átomo, como ya se vio, está integrado por un núcleo formado por un conjunto de protones y neutrones, y por una serie de electrones emplazados a determinada distancia alrededor del núcleo. Electrones, protones y neutrones son las tres partículas básicas. Los electrones poseen una masa muy pequeña y carga negativa. Por su parte, protones y neutrones tienen aproximadamente la misma masa, pero mientras los primeros poseen carga eléctrica positiva, los neutrones carecen de carga. Los electrones del átomo, cuya energía depende de su distancia al núcleo, pueden encontrarse en estado excitado (con una energía superior a la normal) o en reposo. En el estado excitado, el electrón almacena una determinada proporción de energía.

En virtud del llamado proceso de absorción, cuando un fotón (las ondas de luz también se denominan fotones) choca con un electrón no excitado, puede hacer que pase al estado de excitado. Habitualmente, un electrón que resulta excitado, al cabo de un tiempo, pasa nuevamente al estado de reposo por vía de la emisión de un fotón. Este fenómeno, conocido como emisión espontánea, es el que tiene lugar, por ejemplo, en las bombillas eléctricas. Ahora bien, un electrón puede ser inducido a liberar su energía almacenada. Si un fotón pasa al lado de un electrón excitado, este retorna al estado no excitado a través de la emisión de un fotón de luz igual al que pasó junto a él inicialmente. Este proceso se conoce como emisión estimulada y constituye el fundamento del láser.

El rayo láser se diferencia de la luz del sol o de la generada por una bombilla, porque es un haz de luz monodireccional y monocromático.

Los emisores de luz despiden millones de ondas, que pueden tener idéntica dirección o poseer direcciones distintas. La bombilla es un emisor de luz omnidireccional, frente al láser, que es monodireccional. En cuanto a la característica del

monocromatismo, el color de la luz está en función de su frecuencia; si todas las ondas poseen la misma frecuencia, poseen también el mismo color. Los filamentos de las bombillas están formados por átomos y moléculas diferentes y, por tanto, la energía absorbida y desprendida en forma de fotones adopta valores diversos. Puesto que la frecuencia del fotón está en relación con su energía, al variar la energía, varía la frecuencia emitida. La luz de una bombilla tiene múltiples frecuencias, dependiendo del filamento que se haya empleado en su construcción. Por el contrario, en un láser, la fuente de luz proviene de un gas o de un sólido muy purificado. En ambos casos, los átomos tienen idénticos niveles energéticos. Como resultado, los fotones generados poseen idéntica energía y frecuencia.

El láser está formado por un núcleo, que suele tener forma alargada, donde se generan los fotones (Ver Figura 33). El núcleo puede ser una estructura cristalina (por ejemplo, el rubí) o un tubo de vidrio que contiene gases por lo general, dióxido de carbono o la mezcla helio-neón. En cualquier caso, son materiales que poseen electrones fácilmente excitables y que no emiten inmediatamente de forma espontánea, sino que pueden quedar excitados durante un tiempo mínimo. Es precisamente este pequeño intervalo de tiempo el que se necesita para que los electrones produzcan emisión estimulada no espontánea.

Junto al núcleo se halla el excitador, un elemento capaz de provocar la excitación de electrones del material que se halla en el núcleo, a partir de una lámpara de destellos como el flash, semejante al de una cámara fotográfica o de dos electrodos que producen una descarga eléctrica de alta tensión.

El tercer componente del láser son dos espejos paralelos emplazados en los extremos del núcleo. Uno de ellos es reflectante, mientras que el segundo es semirreflectante, es decir, permite el paso de una parte de la luz que le llega. Cuando se verifica la excitación, una gran cantidad de electrones pasan al estado excitado y, una gran mayoría, permanece en dicha situación durante un determinado intervalo de tiempo. No obstante, algunos realizan una emisión espontánea, generando fotones que se desplazan en todas direcciones. Aunque en su mayoría se pierden por los laterales donde no hay espejos, un pequeño número rebota entre ellos y pasa por el interior del núcleo, que es transparente. Al pasar por el núcleo, provocan la emisión estimulada de nuevos fotones en la misma dirección. Estos nuevos fotones rebotan también en los espejos, originando, a su vez, la emisión de más fotones, y así sucesivamente. Puesto que uno de los espejos es semirreflectante, una parte de los fotones, en lugar de rebotar, escapa, formando una especie de chorro muy fino: es el rayo láser visible.

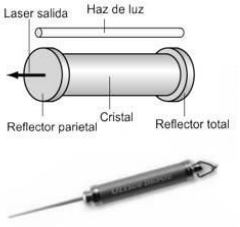


Figura 33

18-Onda electromagnética

Se realizará una breve explicación de lo que son las ondas electromagnéticas, ya que estas son las que se utilizan para las trasmisiones radiales que hoy son un medio habitual de ingreso de datos a las computadoras.

Una onda electromagnética es un tipo de radiación en forma de onda que se caracteriza por poseer dos campos: un campo eléctrico y otro campo magnético, oscilando perpendicularmente entre sí (Ver Figura 34).

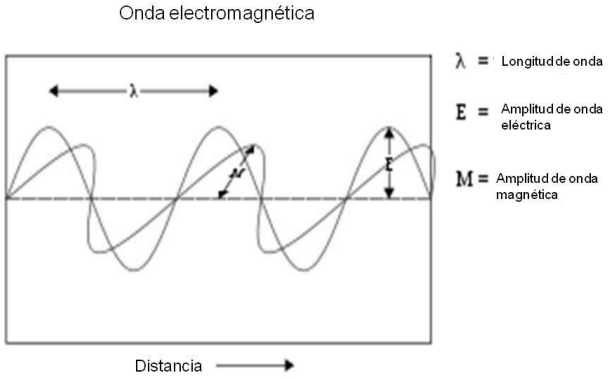


Figura 34

Siguen a continuación una serie de conceptos que es necesario repasar para entender mejor el comportamiento de estas ondas y sus aplicaciones.

Ciclo: se denomina ciclo a cada patrón repetitivo de una onda.

Período: es el tiempo que tarda la onda en completar un ciclo.

Frecuencia: número de ciclos que completa la onda en un intervalo de tiempo. Si dicho intervalo es de un segundo, la unidad de frecuencia es el Hertz (Hz).

Amplitud: es la medida de la magnitud de la máxima perturbación del medio producida por la onda.

Longitud: la longitud de una onda está determinada por la distancia entre los puntos: inicial y final de un ciclo (por ejemplo, entre un valle de la onda y el siguiente). Habitualmente se denota con la letra griega lambda (Λ).

Velocidad: las ondas se desplazan a una velocidad que depende de la naturaleza de la onda y del medio por el cual se mueven. En el caso de la luz, por ejemplo, la velocidad en el vacío se denota "c" y vale 299.792.458 m/s (aproximadamente 3.10^8 m/s).

Fase: la fase de una onda relaciona la posición de una característica específica del ciclo (por ejemplo, un pico), con la posición de la misma característica en otra onda. Puede medirse en unidades de tiempo, distancia, fracción de la longitud de onda o (más comúnmente) como un ángulo.

Polarización: la polarización representa la orientación de cómo la onda oscila, y en el caso particular de las ondas electromagnéticas, la orientación en la oscilación del campo eléctrico. A menudo esta orientación es una línea y por ello se habla típicamente de ondas con polarización vertical u horizontal, es decir, cuando el campo eléctrico oscila en un plano con esas direcciones.

Concepto de modulación

Cuando comparamos el rango de frecuencia típico de la voz humana (400 Hz a 4000 Hz) con el rango de frecuencia de las ondas de radio (a partir de los 30 kHz, aproximadamente), inmediatamente nos damos cuenta de que no es posible convertir directamente de sonido a radio. Es necesario llevar a cabo un proceso intermedio para transmitir una onda de baja frecuencia utilizando una de mayor frecuencia.

Definimos entonces la Modulación como el proceso de alterar las características de una onda (llamada portadora o *carrier*) para que transporte información.

Son varios los parámetros de la onda portadora que podemos alterar, pero los más habituales son la amplitud y la frecuencia (AM y FM).

AM: en este caso, se modifica la amplitud de la portadora en proporción directa a la señal moduladora. Este fue el primer método para la emisión de radio comercial. En la figura 35, se ve en la parte superior una onda de radio (portadora o *carrier*) a la cual se le quiere modular la onda graficada en su interior (*signal*). El resultado de la modulación es el gráfico inferior (*output*).

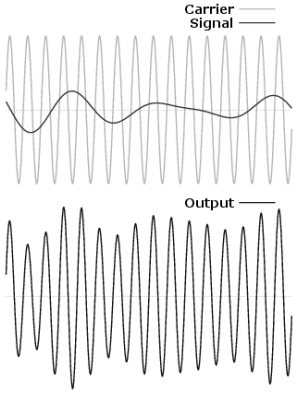


Figura 35

FM: en esta forma de modulación, la información se representa mediante variaciones de la frecuencia instantánea de la onda portadora (Ver Figura 36).

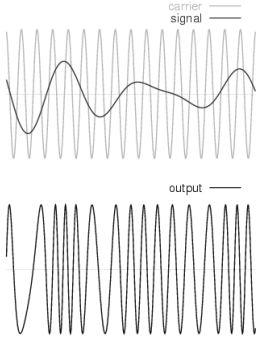


Figura 36

19-Espectro electromagnético

Se denomina espectro electromagnético a todo el rango posible de radiación electromagnética. Esto incluye las ondas de radio, los infrarrojos, la luz, los ultravioletas, los rayos X, gamma, etcétera.

En función de lo anterior, el espectro radioeléctrico o de Radio Frecuencia (RF) se refiere a la porción del espectro electromagnético en el cual las ondas electromagnéticas pueden generarse alimentando a una antena con corriente alterna.

La tabla a continuación presenta las bandas de RF más importantes:

Abreviatura	Nombre	Frecuencia
VLF	Very Low Frequency	3-30 kHz
LF	Low Frequency	30-300 kHz
MF	Medium Frequency	300-3000 kHz
HF	High Frequency	3-30 MHz
VHF	Very High Frequency	30-300 MHz
UHF	Ultra High Frequency	300-3000 MHz
SHF	Super High Frequency	3-30 GHz
EHF	Extremely High Frequency	30-300 GHz

20-Los osciladores

El oscilador es el instrumento utilizado para generar señales de radiofrecuencia, que pueden estar comprendidas entre los 100 kHz y los 1000 MHz. Las señales que se utilizan para este tipo de transmisión son senoidales (Ver Figura 37).

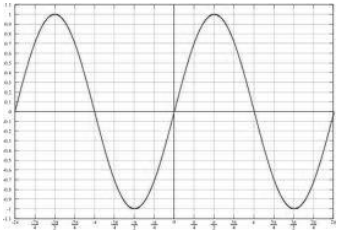


Figura 37

Para generar estas ondas, el oscilador utiliza circuitos resonantes que se forman con bobinas y condensadores conectados en serie o en paralelo (Ver Figura 38).

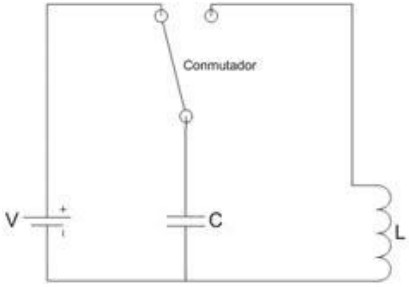


Figura 38

Cuando en el circuito se conecta la fuente con el condensador (V con C), como esta en la imagen, la bobina L dirige electrones hacia el condensador en el sentido de L a C, ya que el condensador se está cargado. Cuando el conmutador une en un circuito al condensador con la bobina (V con L), los electrones circulan por la bobina en sentido inverso, porque en este caso el condensador se está descargando.

Este cambio de sentido sobre la bobina hace que esta genere ondas electromagnéticas de tipo senoidales.

21-El modulador

Un modulador es un aparato que recibe dos señales de entrada de información: una señal portadora de amplitud constante y de frecuencia sencilla (portadora), y la señal de información. La información actúa sobre la portadora a la que modula. Debido a que la información actúa sobre la portadora, se la llama señal modulante. La resultante se llama onda modulada o señal modulada (Ver Figura 39).

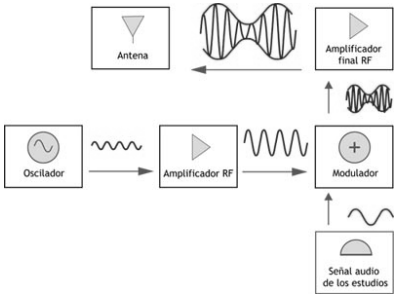


Figura 39

Capítulo III
Esquema y funcionamiento básico de una computadora

El hardware de las computadoras está integrado por circuitos electrónicos formados por condensadores, diodos, transistores y transformadores, entre los componentes más importantes.

El punto primordial es tratar de imaginar cómo una computadora logra efectuar cálculos matemáticos y realizar diferentes acciones de acuerdo a determinados parámetros, como una presión en una tecla, un simple click en algún botón del mouse o una determinada letra en algún lugar de la memoria primaria (RAM). Y además, imaginarlo sabiendo que por adentro de la computadora lo único que se mueven son, básicamente, electrones.

1-Funcionamiento en general

Quienes se pusieron a pensar cómo hacer una máquina (luego llamada computadora), que ayudara al hombre a realizar cálculos matemáticos a gran velocidad, almacenar millones de datos y poder consultarlos rápidamente, primero debieron pensar con qué contaban.

Los ingredientes fundamentales con que contaban los científicos fueron:

- Corriente eléctrica que circula por cables.
- Transformadores de corriente para trabajar con muy poco voltaje.
- La posibilidad de bifurcar la corriente eléctrica utilizando compuertas, como los transistores NMOS o PMOS (esto es en términos de tecnología moderna y no de la que contaban los científicos en los primeros momentos de la computación).
- La posibilidad de almacenar corriente eléctrica en condensadores o transistores.

Con estos ingredientes, se intentaron imaginar el funcionamiento de las computadoras modernas.

Los condensadores parecían ser una buena herramienta, pueden estar con carga eléctrica o sin ella. Esto en primera instancia ya parece una memoria. Algo que queda en el interior de la computadora, al menos por un tiempo. Parecido a las cosas que quedan en el cerebro de los seres humanos, algunas son por un corto tiempo y otras durante gran parte de la vida. El problema está en que en nuestra cabeza quedan (no sabemos muy bien cómo) simbolismos que los humanos sabemos

reconocer: número, palabras, imágenes, olores, etcétera. Bastante distinto a recordar, solamente, si hay energía o no.

Pero para los científicos, esto pareció ser un buen comienzo. Pensaron que en lugar de manejarse con una numeración decimal, o un alfabeto complejo (como lo hacemos los humanos), se podrían manejar con una codificación binaria, estos es: de dos estados, tiene energía o no tiene energía.

A estos dos estados se los suele representar como 0 y 1. A esta variable de 0 o 1, se la denomina bit. Bit es el acrónimo de *binary digit* (dígito binario). Como se verá más adelante, el bit también puede ser representado por magnetismo (polaridad positiva o negativa), por presencia de luz o no, etcétera.

Entonces, los científicos pensaron en una tabla de conversión entre un código binario y el simbolismo que interpretan los humanos. Hoy hay varias tablas de conversión, pero se nombrarán las más utilizadas.

1.1-Tablas de conversión de codificación humana a codificación binaria.

EBCDIC (*Extended Binary Coded Decimal Interchange Code*): desarrollado por la empresa IBM, es de 8 bits, es decir, que para representar cada carácter alfanumérico, se necesitarían 8 bits.

ASCII (*American Standard Code for Information Interchange*): fue creando por ANSI (*American National Standards Institute*), es una organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos. También es de 8 bits.

En la figura 40, se puede observar cómo se convierten 8 bits, según cada tabla, en un símbolo que reconocen los humanos. Con estas codificaciones, se representaban hasta 256 caracteres.

Hoy se está tendiendo a unificar la tabla de conversión en el estándar Unicode. El estándar Unicode es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de múltiples lenguajes y disciplinas técnicas. El término Unicode proviene de los tres objetivos perseguidos: universalidad, uniformidad y unicidad. Este estándar es mantenido por el Unicode Technical Committee (UTC), integrado en el Unicode Consortium, del que forman parte con distinto grado de implicación empresas como: Microsoft, IBM, Oracle, SAP, Google, instituciones como la Universidad de Berkeley, y profesionales y académicos a título individual.

Unicode puede requerir 8, 16, 24 o 32 bits. Existen diferentes tablas, la más utilizada es la de 16 bits que implican una capacidad de representar 65.535 caracteres.

Character	EBCDIC Binary	Character	ASCII-8-Binary
A	1100 0001	A	1010 0001
B	1100 0010	B	1010 0010
C	1100 0011	C	1010 0011
D	1100 0100	D	1010 0100
E	1100 0101	E	1010 0101
F	1100 0110	F	1010 0110
G	1100 0111	G	1010 0111
H	1100 1000	H	1010 1000
I	1100 1001	I	1010 1001
J	1101 0001	J	1010 1010
K	1101 0010	K	1010 1011
L	1101 0011	L	1010 1100
M	1101 0100	M	1010 1101
N	1101 0101	N	1010 1110
O	1101 0110	O	1010 1111
P	1101 0111	P	1011 0000
Q	1101 1000	Q	1011 0001
R	1101 1001	R	1011 0010
S	1110 0010	S	1011 0011
T	1110 0011	T	1011 0100
U	1110 0100	U	1011 0101
V	1110 0101	V	1011 0110
W	1110 0110	W	1011 0111
X	1110 0111	X	1011 1000
Y	1110 1000	Y	1011 1001
Z	1110 1001	Z	1011 1010
0	1111 0000	0	0101 0000
1	1111 0001	1	0101 0001
2	1111 0010	2	0101 0010
3	1111 0011	3	0101 0011
4	1111 0100	4	0101 0100
5	1111 0101	5	0101 0101
6	1111 0110	6	0101 0110
7	1111 0111	7	0101 0111
8	1111 1000	8	0101 1000
9	1111 1001	9	0101 1001

Figura 40

Como base para calcular datos de capacidades, tanto de memoria primaria como de soportes o almacenamientos externos, se utiliza la base de 8 bits a la que se denomina byte.

Por lo tanto 1 byte son 8 bit. Y así surge la tabla de medida en byte:

1 Byte	= 8 bits
1 Kilobyte (KB)	= 1000 Bytes
1 Megabyte (MB)	= 1000 KB
1 Gigabyte (GB)	= 1000 MB

1 Terabyte (TB)	= 1000 GB
1 PetaByte (PB)	= 1000 TB
1 ExaByte (EB)	= 1000 PB
1 ZettaByte (ZB)	= 1000 EB
1 YottaByte (YB)	= 1000 ZB
1 XeraByte (XB)	= 1000 YB

En algunas bibliografías, aún se mantiene la idea de que un 1 KB es igual a 1024 bytes y no a 1000 bytes. Esto ocurre porque originalmente las memorias primarias de las computadoras o memorias RAM siempre han sido matrices cuadradas. En un circuito de memorias, por cuestiones técnicas, siempre se han utilizado memorias de 8 bits x 8 bits o de 16 bits x 16 bits o de 32 bits x 32 bits, y así han ido creciendo. Esto significaba que una memoria nunca podía tener exactamente 1000 bytes, ya que una matriz cuadrada de 1000 bytes no es un número entero. En cambio, una matriz cuadrada de 1024 bytes (1 KB) sería de 32 bytes x 32 bytes.

Ahora, para que se entienda un poco, se partirá de una pregunta: ¿cuántas letras F entrarían en 1 MB utilizando la codificación ASCII (recordemos que para codificar en ASCII un carácter se requiere de 1 byte)? Si 1 MB son 1000 KB y 1 KB son 1000 bytes por lo tanto entran 1.000.000 letras F.

La numeración binaria permite realizar cálculos matemáticos como con cualquier otro sistema (octal, decimal, hexadecimal, etcétera.)

En ASCII:

- El 0 decimal es el 0101 0000 binario.
- El 1 decimal es el 0101 0001 binario
- El 2 decimal es el 0101 0010 binario
- El 3 decimal es el 0101 0011 binario
- El 4 decimal es el 0101 0100 binario
- El 5 decimal es el 0101 0101 binario
- El 6 decimal es el 0101 0110 binario
- El 7 decimal es el 0101 0111 binario
- El 8 decimal es el 0101 1000 binario
- El 9 decimal es el 0101 1001 binario

Como se puede ver, ASCII construye los caracteres numéricos manteniendo los primeros cuatro bits fijos en 0101 y luego numera binariamente los siguientes cuatro bits.

El 0 decimal es igual al 0000 binario y el 1 decimal es igual al 0001 binario. El número decimal 2 en binario es 0010. Como se puede ver, el primer dígito a la derecha

vuelve a 0 y se incrementa en 1 el segundo dígito. Luego, el 3 decimal se logra aumentando en 1 el primer dígito, y queda el número en 0011 binario. El 4 decimal es igual a 100 binario, ya que al incrementar 1 el primer dígito, se tiene que volver a cero; luego, se debería incrementar el segundo, pero como ya es un 1, se debe volver a 0 y luego incrementar en 1 el tercer dígito. El 5 decimal es 0101 binario, ya que se debe incrementar en 1 el primer dígito. El 6 decimal es el 0110 binario, ya que al incrementar en 1 el primer dígito, se debe volver a 0 y sumar 1 al segundo dígito. El 7 decimal es el 0111, ya que se debe sumar 1 al primer dígito. El 8 decimal es el 1000 binario, ya que al sumar un 1 al primer dígito, se debe volver a 0, al incrementar en 1 el segundo dígito se debe volver a 0, al incrementar en 1 el tercer dígito debe volverse a 0 y luego incrementar en 1 el cuarto dígito. El 9 decimal es el 1001 binario, ya que se debe incrementar en 1 el primer dígito.

1.2-El cálculo matemático en circuitos electrónicos

En la numeración binaria, es posible realizar los mismos cálculos matemáticos que con el sistema decimal. Pero como el objetivo de estos párrafos es que se llegue a comprender cómo funcionan los circuitos electrónicos de la computadora, bastará con realizar una demostración sobre la suma.

Como sabemos, 2 + 3 es igual a 5. Ahora, en el sistema binario de la tabla ASCII, debería ser que 0101 0010 (2) + 0101 0011 (3) debe ser igual a 0101 0101 (5).

En la suma de números, los primeros cuatro binarios, 0101, se descartan o se reutilizan, pero para el ejemplo se considerará que se los descarta. Se debería sumar 0010 + 0011 (Ver Figura 41).

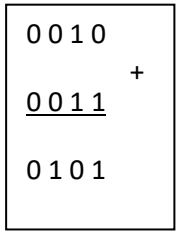


Figura 41

La suma en el sistema binario es muy simple:

1) Se comienza por la derecha y se suma 1 + 0 que es igual a 1 y está dentro de los valores binarios posibles. Por lo tanto, queda el 1.

2) Se toma la segunda hilera a partir de la derecha y se tiene que sumar 1 + 1 que en decimal sería igual a 2, pero como 2 ya no es un número admitido por el sistema binario, se coloca un 0 como resultado y se acarrea un 1 a la tercera columna.
3) En la tercera columna, se tendría que sumar el 1 del acarreo + 0 + 0 lo que sería igual a 1. Por lo tanto, se coloca un 1 en la tercera columna.
4) En la cuarta columna, se tiene 0 + 0 que es igual a 0. Por lo tanto, se coloca un 0 como resultado de dicha suma.

Ahora se debe demostrar cómo hace la computadora este cálculo con circuitos electrónicos. Para este fin, se utiliza el álgebra proposicional.

1.3-Algebra proposicional o compuertas lógicas

Las proposiciones son enunciados declarativos que afirman o niegan algo. Por ende, las proposiciones pueden ser verdaderas o falsas.

Ahora bien, las proposiciones pueden combinarse de tres formas básicas: AND, OR, XOR. Existen otras formas, pero que son derivaciones de estas. Las proposicionales, también llamadas compuertas lógicas, son la técnica en que se fundamentan las computadoras para poder realizar operaciones matemáticas. Se las denominan compuertas, ya que de determinado input permiten, abriendo o cerrando compuertas, un determinado output.

Hay una compuerta lógica, adicional, muy importante para la construcción de los circuitos computacionales que se denomina NOT. Esta compuerta no es tanto una proposición, sino que es más bien una inversión. Si la entrada a la compuerta es verdadera, la salida es falsa; y si la entrada es falsa, la salida es verdadera.

Proposiciones básicas:

Conjuntiva (AND):

La proposición AND exige que las dos premisas sean verdaderas para que la conclusión sea verdadera.

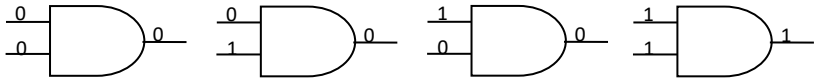
De esta lógica, surge la siguiente tabla de verdad:

1P	2P	C
F	F	F
F	V	F
V	F	F
V	V	V

Como se puede ver, esta es una tabla binaria que puede ser representada por unos y ceros. El 1 es considerado Verdadero y el 0 es considerado Falso. Esta forma de representación se llama lógica positiva y se representa de la siguiente forma:

1P	2P	C
0	0	0
0	1	0
1	0	0
1	1	1

Para graficar estas compuertas, se utiliza el siguiente símbolo:



Disyuntiva (OR):

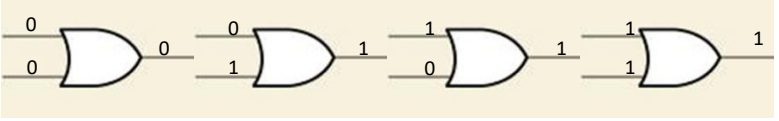
En este caso, si alguna de las premisas es verdadera o las dos son verdaderas, la conclusión puede ser verdadera. Solo si las dos premisas son falsas, la conclusión nunca podrá ser verdadera.

De esta lógica, surge la siguiente tabla de verdad:

1P	2P	C
F	F	F
F	V	V
V	F	V
V	V	V

1P	2P	C
0	0	0
0	1	1
1	0	1
1	1	1

Para graficar esta lógica proposicional, se utiliza el siguiente símbolo:



Disyuntiva exclusiva (XOR)

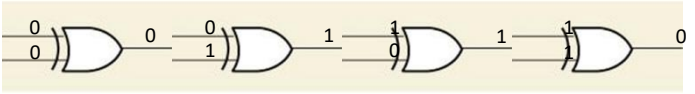
La disyuntiva exclusiva implica que la conclusión será verdadera exclusivamente si se da una de las dos condiciones, debido a que las dos juntas no pueden darse nunca.

De esta lógica, surge la siguiente tabla de verdad:

1P	2P	C
F	F	F
F	V	V
V	F	V
V	V	F

1P	2P	C
0	0	0
0	1	1
1	0	1
1	1	0

Para dibujar esta lógica proposicional, se utiliza el siguiente símbolo:



Compuerta NOT

La salida es una inversión de la entrada.

De esta lógica, surge la siguiente tabla:

1P	Inv.
F	V
V	F

1P	Inv.
0	1
1	0

Para graficar esta compuerta, se utiliza el siguiente símbolo:

1.4-Circuito sumador con simbología proposicional

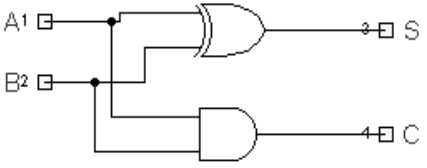
Se comenzará con el diseño de un sumador de 1 bit. Para ello se utilizará una combinación proposicional de XOR y AND.

En el siguiente ejemplo (Ver Figura 40), se muestran las cuatro combinaciones posibles de sumar un número binario. El resultado de la Suma entre 0+0=0, entre 0+1=1, entre 1+0=1 y entre 1+1=, pero se debe acarrear uno a la izquierda.

Bit A	Bit B	Suma	Acarreo
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figura 42

El circuito en lógica proposicional del sumador de 1 bit (semisumador) queda de la siguiente forma:



Imagínese, estimado lector, que se está ingresando 0 volt por el cable de cobre A y 0 volt por el cable de cobre B. La compuerta lógica XOR recibirá en ambos casos O volts, por lo tanto, en la salida se recibirá 0 volt. La compuerta AND recibirá también en ambos casos 0 volt, por lo tanto, la salida del cable de cobre de acarreo (C) será de 0 volt.

Pero yendo al caso más interesante que es el de la última fila (1+1=1), se puede ver cómo funciona: la compuerta lógica XOR recibirá en ambos casos 5 volt, por lo que la salida por el cable de cobre será de 0 volt. En la compuerta lógica AND también se están recibiendo 5 volt en ambos casos, por lo tanto la salida por el cable de cobre del acarreo (C) será de 5 volt. Cabe aclarar aquí que cuando se ve la implementación física del modelo, se observará que el 0 binario puede ser un rango de entre 0 y 1 volt, mientras que el 1 binario puede ser de entre 3 volt y 5 volt (según el tipo de transistor que se utilice).

Ahora ya estamos en condiciones de desarrollar el sumador completo de 1 bit considerando el acarreo. Utilizaremos el ejemplo que queríamos sumar de 4 bits para ir viendo el resultado final:

Bit A	Bit B	Acarreo de inicio (CI)	Suma (S)	Acarreo de salida (CS)
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
0	0	0	0	0

Le sugiero al lector que pruebe cada una de las filas de la tabla en el circuito y verá que los resultados de S, CI y CS son correctos (Ver Figura 43).

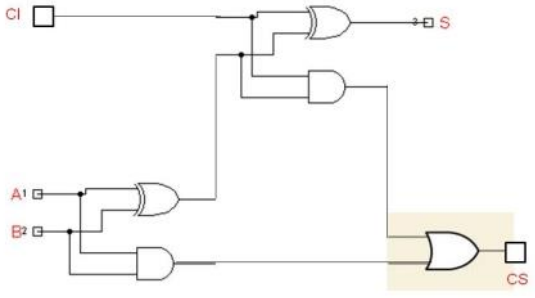
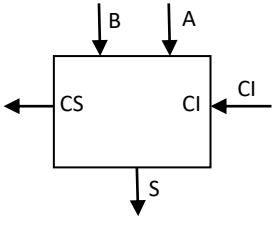
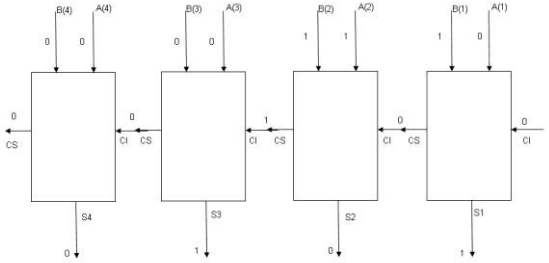


Figura 43

Ahora se desarrollará un sumador de 4 bits. Para lo cual se empaquetará el sumador de 1 bit en un pequeño circuito:



Para por último construir el circuito sumador de 4 bits:



Para comprender con más claridad todo este andamiaje tecnológico, faltará desarrollar el paso de estas compuertas lógicas a una construcción física, y aquí se verá que para cada una de estas compuertas lógicas se requieren varias compuertas físicas que se implementan mediante la utilización de los transistores.

1.5-Una compuerta lógica AND implementada con transistores NMOS y PMOS

La implementación de la compuerta AND con los transistores NMOS y PMOS se implementan con una negación de la función lógica AND (NOT AND o también llamada NAND) y luego con una función NOT o inversora que nos permite obtener el AND limpio.

Es muy importante recordar que los transistores NMOS abren las compuertas recibiendo electrones (carga negativa) y que los transistores PMOS abren las compuertas recibiendo carga positiva (Ver Figura 44).

Este circuito debería responder a la siguiente tabla de verdad:

1.er Bit	2.do Bit	Bit de salida
0	0	1
0	1	1
1	0	1
1	1	0

Si al bit de salida se introduce en un circuito inversor, quedaría un AND.

Para entenderse el circuito, es conveniente que se realice el seguimiento de cada combinación posible. Luego de realizar el circuito, el lector se dará cuenta de que los transistores tipo PMOS serán suficiente para dar por salida la respuesta necesaria. El circuito se implementa con el agregado de dos transistores tipo NMOS, al solo efecto de evitar tener que usar resistencias y aprovechar los transistores como resistencias. Se implementa de esta forma porque los transistores son mucho más pequeños que las resistencias.

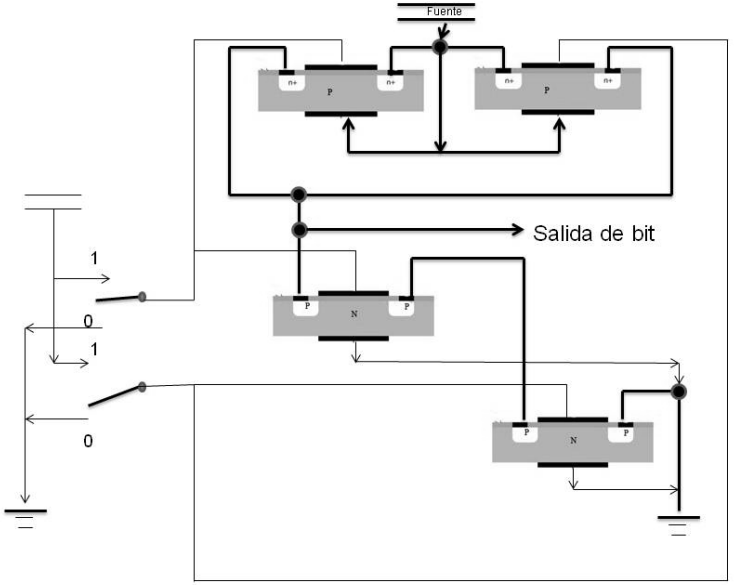


Figura 44

2-Diagrama básico de unidades funcionales de la computadora (Figura 45)

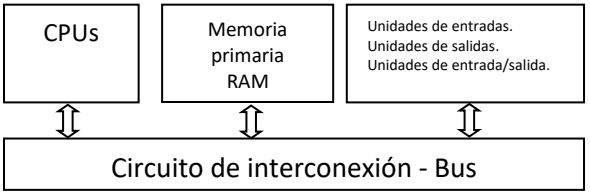


Figura 45

Se realizará una breve descripción de cada unidad funcional, ya que (sin querer desmerecer para nada el magnífico desarrollo tecnológico), básicamente son circuitos, cada vez más pequeños, cada vez más veloces y que cada vez permiten manejar mayor volumen de información.

2.1-La CPU (Ver Figura 46)

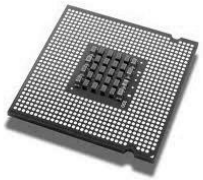


Figura 46

La CPU, así denominada por el acrónimo en inglés de *Central Processing Unit*, son las unidades centrales de procesamiento, también llamadas comúnmente microprocesadores. Es el componente del computador que interpreta las instrucciones contenidas en los programas y que procesa los datos. La operación fundamental de la CPU es ejecutar una secuencia de instrucciones almacenadas llamadas «programa». La CPU está integrada por la unidad de aritmética/lógica (ALU) y la unidad de control (CU). Una computadora puede contener desde una CPU hasta cientos de ellas.

> Unidad aritmética y lógica: realiza todos los cálculos, todas las comparaciones y genera los resultados. Cuando la unidad de control encuentra una instrucción de aritmética o de lógica, le envía el control a la unidad de aritmética y lógica.
>
> La unidad de control: coordina y gobierna todas las operaciones que se realizan en la
> CPU. Comprueba y administra las demás partes de la computadora. Además, selecciona, verifica e interpreta las instrucciones del programa y después verifica que se ejecuten

Contiene los siguiente subcircuitos:

- Contador de programa: su tarea es tomar de la memoria la instrucción que será ejecutada y apuntar a la siguiente instrucción por ejecutar.

- Registro de instrucción: almacena la instrucción que se está realizando.
- Decodificador: extrae el código de instrucción del registro de instrucción, lo interpreta y envía la información necesaria para su ejecución al secuenciador.
- Secuenciador: genera órdenes elementales para ser enviadas a la UAL, a la RAM o a los controladores de entrada-salida.
- Reloj: es el circuito que va generando los pulsos para que se ejecuten las instrucciones. Es el que imprime la velocidad de la CPU.
- Memorias cache L1, L2, y L3 (se explicará luego).

La instrucción que ejecuta el secuenciador está expresada en forma binaria, es decir, unos y ceros. Por ejemplo, la instrucción de poner una variable (o espacio de memoria) llamada «contador» en el valor cero, sería algo muy parecido a lo siguiente: 00000111010100011111000010001000010101010. Y la instrucción será diferente según el tipo y la marca de procesador que se trate.

Existen dos familias de procesadores: los CISC (*Complex Instruction Set Computer*) en español es «set de instrucciones complejas», y los RISC (*Reduced Instruction Set Computer*) en español, «set de instrucciones reducidas». La diferencia entre ellos radica en que los procesadores CICS utilizan un importante conjunto de instrucciones complejas. La instrucción de programa CICS es interpretada o, se podría decir, desmenuzada por un microprograma en subinstrucciones. El microprograma de estos microprocesadores está grabado en una memoria ROM (grabada de fábrica e inalterable). Esto hace más lenta la corrida del programa, ya que se requiere de un esfuerzo de traducción antes de que el microprocesador pueda ejecutarla. Los procesadores RISC utilizan un set mucho más pequeño de instrucciones simples. Las funcionalidades a las que se llega tanto del CICS como del RISC son las mismas, lo que ocurre es que los RISC utilizan muchas instrucciones simples que equivalen a una compleja del CICS. Por ende, los programas para procesadores RISC son mucho más largos, pero no requieren de un microprograma que interprete la instrucción. El código utilizado por el microprocesador RISC es una instrucción binaria que está lista para ser ejecutada, esto hace que la instrucción se ejecute muy rápido, pero a la vez, se ejecutan muchas más instrucciones que en los SICS.

Velocidad de la CPU: Las computadora operan por medio de pulsos eléctricos, como el tic tac de un reloj. Tic, la computadora toma una instrucción de la memoria; toc, se recupera un dato; tic, se suma a otro dato; toc, se guarda en la memoria, etcétera. Cada uno de estos pulsos configura un ciclo dentro de la CPU. Mientras más pulsos se puedan completar en determinado tiempo, más rápida será la computadora. Hertz (Hz) es una medida de «ciclos por segundo», o sea, cuántas veces pasa algo en un segundo. En este caso, se refiere a los pulsos de la computadora, es decir, cuántas operaciones hace por segundo. Para su expresión, se siguen las convenciones del

sistema internacional: 1 kHz = 1000 Hz, 1 GHz = 1.000.000.000 Hz. Así que una computadora de 3 GHz está haciendo tres mil millones de ciclos por segundos.

Básicamente, la memoria caché de un procesador es un tipo de memoria volátil (del tipo RAM), pero de una gran velocidad. En la actualidad, esta memoria está integrada en el procesador, y su cometido es almacenar una serie de instrucciones y datos a los que el procesador accede continuamente, con la finalidad de que estos accesos sean instantáneos. Estas instrucciones y datos son aquellos a los que el procesador necesita acceder de forma continua, por lo que para el rendimiento del procesador es imprescindible que este acceso sea lo más rápido y fluido posible. Hay, básicamente, tres tipos diferentes de memoria caché para procesadores:

- Caché de 1.er nivel (L1): esta caché está integrada en el núcleo del procesador, trabajando a la misma velocidad que este. La cantidad de memoria caché L1 varía de un procesador a otro, y está normalmente entre los 64 KB y los 256 KB. Esta memoria suele a su vez estar dividida en dos partes, una dedicada para instrucciones y otra para datos.

- Caché de 2.do nivel (L2): integrada también en el procesador, aunque no directamente en el núcleo de este, tiene las mismas ventajas que la caché L1, aunque es algo más lenta que esta. La caché L2 suele ser mayor que la caché L1, y puede llegar a superar los 2 MB. A diferencia de la caché L1, esta no está dividida, y su utilización está más encaminada a programas que al sistema.

  Caché de 3.er nivel (L3): es un tipo de memoria caché más lenta que la L2, suele ser la más alejada del núcleo. Prácticamente tiene las mismas funciones que la L2. Puede tener una capacidad de hasta 8 MB o más.

2.2-La memoria primaria o RAM (Ver Figura 47)

Figura 47

RAM es el acrónimo del las palabras en inglés *Random Access Memory*. Es un memoria a la que la CPU puede acceder aleatoriamente; es decir, se puede acceder a cualquier byte de memoria sin acceder a los bytes precedentes (no es necesario recorrerla toda o parcialmente hasta encontrar el dato). Una de las características fundamentales es que es una memoria que solo se conserva mientras la computadora está encendida y conectada a la fuente de energía. La CPU solo puede ejecutar instrucciones u operar datos que se encuentren ingresados a la memoria RAM.

Las actuales computadoras utilizan unas memorias denominadas SDRAM (*Synchronous Dynamic Random Access Memory*). La característica más importante de estas memorias es que cada celda de memoria (cada bit) está formado por un condensador que es cargado o descargado mediante la utilización de un transistor EMOS que cumple la función de switch para cargar o descargar el condensador (Ver Figura 48).

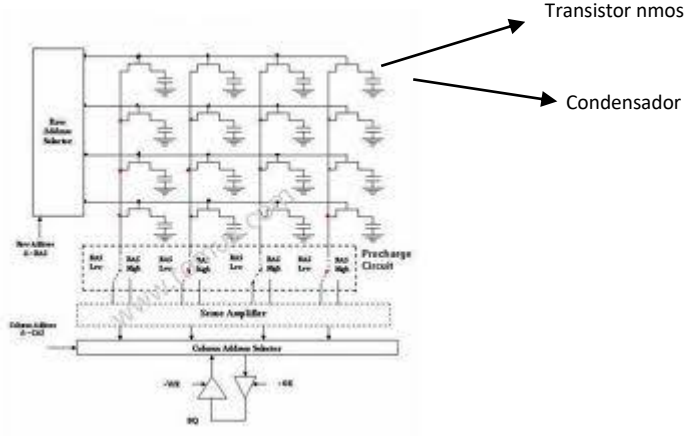


Figura 48

Cada una de las celdas o intersección de fila y columna está formada por un transistor y un condensador. Si el condensador tiene un voltaje alto, este representa un bit en 1; y si el condensador tiene un voltaje bajo, este representa un bit en 0.

En la figura 49, se observa lo que sería una celda o un bit de memoria.

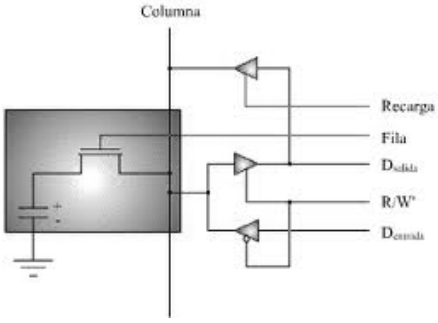


Figura 49

Una explicación simplificada de la carga y descarga del condensador o bit

Carga del condensador: si la línea de Fila envía corriente para abrir el transistor NMOS, y simultáneamente se envía corriente por la Columna, desde la línea de de entrada, se podrá cargar el condensador.

Recarga del condensador: cada un periodo de plazo muy pequeño, el condensador comienza a perder la energía; por lo tanto, antes de que la pierda, debe ser recargado. Esto se realiza con la línea de Recarga enviando energía, y la línea de Fila permitiendo abrir el transistor.

Lectura del condensador: abriendo el transistor con corriente por la Fila, dejando la Columna sin energía para que la línea de salida pueda tomar el valor de energía que tiene el condensador. Inmediatamente después de la lectura, es necesaria una recarga.

El borrado del condensador: sin recargar el condensador durante el periodo de tiempo requerido, el condensador queda borrado o sin energía.

2.3-El bus

Son los canales de comunicación que une los componentes anteriormente enunciados. La función del bus es la de permitir la conexión entre distintos subsistemas de un sistema digital, enviando datos entre dispositivos muy variados. La mayoría de los buses están basados en conductores metálicos por los cuales se trasmiten señales eléctricas que son enviadas y recibidas por los distintos dispositivos de una computadora. Las señales digitales que se trasmiten son de datos, de direcciones o señales de control. Los buses definen su capacidad de acuerdo a la frecuencia máxima de envío y al ancho de los datos.

Existen una gran variedad de buses que se utilizan conjuntamente en una misma computadora. Genéricamente, se los puede clasificar de dos formas: paralelo y serie.

Bus paralelo

Es un bus en el cual los datos son enviados por bytes al mismo tiempo, con la ayuda de varias líneas que tienen funciones fijas. Por ejemplo, el *front-sid*e *bus* de los procesadores Intel es un bus de este tipo y, como cualquier bus, presenta unas funciones en líneas dedicadas:

Las líneas de dirección son las encargadas de indicar la posición de memoria o el dispositivo con el que se desea establecer comunicación.

Las líneas de control son las encargadas de enviar señales de arbitraje entre los dispositivos.

Las líneas de datos trasmiten los bits de forma aleatoria

Un bus paralelo tiene conexiones físicas complejas, pero la lógica es sencilla: lo hace útil en sistemas con poco poder de cómputo.

Bus serie

En este tipo de bus los datos son enviados, bit a bit y se reconstruyen por medio de registros o rutinas de software. Está formado por pocos conductores y su ancho de banda depende de la frecuencia. Es usado desde hace menos de diez años en buses para discos duros, unidades de estado sólido, tarjetas de expansión y para el bus del procesador.

Cada tipo de bus tiene sus características particulares y en general un tipo especial de conector. Los conectores sirven para unir el bus a un periférico o a una Interfaz con otro bus. Es decir, que cuando dos buses de diferentes características se unen, es necesario intercalar una Interfaz (un circuito electrónico) que regule la comunicación y sirva de adaptador físico.

A continuación se describirá cómo se conectan las diferentes unidades de un computador en la mayoría de las computadoras de hoy en día (Ver Figura 50).

Las CPU y las memorias RAM, por la necesidad de estar altamente integradas, se conectan muy fuertemente mediante un puerto de alta velocidad serial. Es muy posible que a este bus se conecten los monitores.

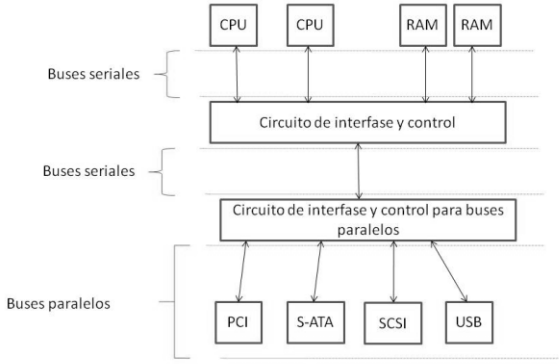


Figura 50

En los buses se trasmite información digital mediante el uso de corriente eléctrica o señales electrónicas.

Las señales electrónicas pueden dividirse en dos grandes grupos: las señales analógicas y las señales digitales.

Señales analógicas: una señal analógica puede verse como una forma de onda que toma un continuo de valores en cualquier momento dentro de un intervalo de tiempo. Por ejemplo, utilizando un micrófono, es posible pasar la onda sonora de la voz humana a una pequeña señal eléctrica, donde el nivel de tensión siga una analógica con la variación sonora en volumen y frecuencia (Ver Figura 51).

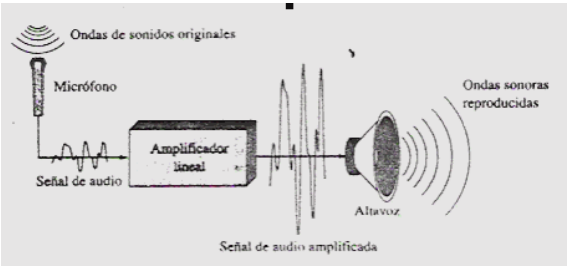


Figura 51

Señales digitales: la señal electrónica digital es muy simple, ya que su objetivo es trasmitir dos tipos de estados, 1 o 0, alto o bajo o como quiera designarse a los estados binarios.

Por ejemplo, las señales eléctricas analógicas de la gráfica anterior pueden ser digitalizadas tomando muestras del valor de la señal y pasando ese valor a un valor binario, y luego en lugar de trasmitir una señal analógica, se trasmite una señal digital (Ver Figura 50). En esta imagen, se puede ver, en la parte superior, una onda analógica, y en la parte inferior, los puntos de muestreo que serán tomados de esa onda para luego ser digitalizados. La onda digital trasmitirá, por ejemplo, 8 bits para expresar cada punto del muestreo seleccionado.

En la figura 51, se puede ver en la parte inferior, una onda digital cuyos valor es alto (que pueden ser entre 3 y 5 volts) representa un 1 y cuyos valores bajos (que pueden ser entre 0 y 1 volt) representan un 0. La figura de la parte superior muestra la representación binaria de la onda inferior. Por ejemplo, los 8 bits de la figura 53 representan tan solo un punto de la muestra de la figura 52 en su parte inferior.

De lo expuesto, se puede observar que cuántas más muestras se tomen de la onda, mayor parecido a la onda original tendrá.

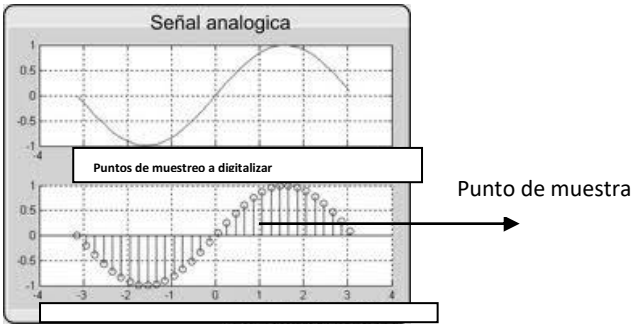


Figura 52

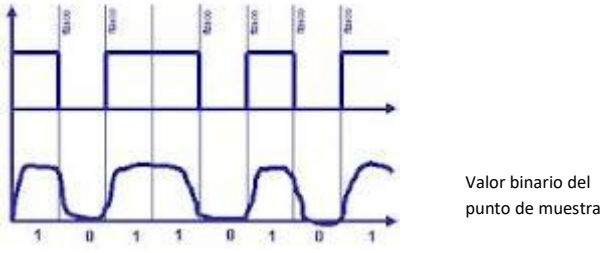


Figura 53

Muchas veces la computadora, para poder manejar las unidades periféricas, se ve obligada a cambiar una señal digital a una señal analógica o viceversa.

En las características de los buses, es muy importante lo que se llama el ancho de palabra, es decir, la capacidad de enviar bits simultáneos. En general, los buses manejan de 16, 32 o 64 bits simultáneamente.

2.4-Buses y periféricos

Se enumerarán los buses más utilizados y los periféricos que suelen unirse a estos buses.

PCI

Interconexión de Componentes Periféricos (PCI, *Peripheral Component Interconnect*). Consiste en un bus de ordenador estándar para conectar dispositivos periféricos. Puede ser de 32 bits y de 64 bits. La conexión a este bus es por medio de un sócalo (Ver Figura 54) que permite la inserción de circuitos integrados para controlar y lograr la adaptación del periférico a la computadora.

Figura 54

Muchos periféricos utilizan este tipo de bus para enlazarse con la computadora. Son muy conocidas las placas de sonido, que dentro de las funcionalidades que presta una placa de sonido, tiene la de convertir la información digital en analógica para que pueda ser trasmitida a amplificadores, auriculares, etcétera.

También son muy utilizadas las placas PCI para comunicaciones de red, tanto para WIFI como para cable (con conectores RJ45).

A su vez, se utiliza para conectar placas controladoras de video, de escáneres, impresoras, etcétera.

S-ATA

Serial ATA o SATA (acrónimo de *Serial Advanced Technology Attachment*) es un bus de transferencia de datos entre la computadora y algunos dispositivos de almacenamiento, como puede ser el disco duro, lectores y regrabadores de CD/DVD/BR, Unidades de Estado Sólido. SATA proporciona la capacidad para conectar unidades al instante, es decir, insertar el dispositivo sin tener que apagar el ordenador o sin que sufra un cortocircuito.

Actualmente, es un bus aceptado y estandarizado. La Organización Internacional Serial ATA (SATA-IO) es el grupo responsable de desarrollar, de manejar y de

conducir las especificaciones estandarizadas de Serial ATA (Ver Figura 55). Estos han reemplazado a los viejos bus IDE.

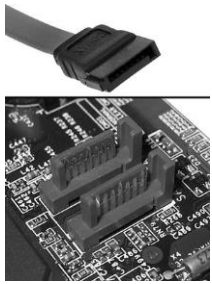


Figura 55

SCSI

SCSI (*Shugart Associates Systems Interface*) es un tipo de bus; la interfaz SCSI, conocida también como adaptador host, adopta la forma de una tarjeta que se inserta en un bus PCI, de la que sale un bus (cable), en el que se pueden conectar varios dispositivos. Este adaptador host es en realidad un puente entre el bus SCSI y el bus PCI.

Desde el punto de vista del Sistema, los dispositivos SCSI son muy eficientes. Soportan comandos del tipo «Rebobina esta cinta» o «Formatea este disco» sin intervención del procesador, con lo que se ahorra tiempo de proceso. Esto es especialmente importante en sistemas multitarea como Unix, Linux, Windows, etcétera.

El bus SCSI es muy flexible y no solo permite conectar discos, también otros periféricos, como escáneres, unidades de cinta, CD-ROM, DVDs, etcétera. (Ver Figura 56).

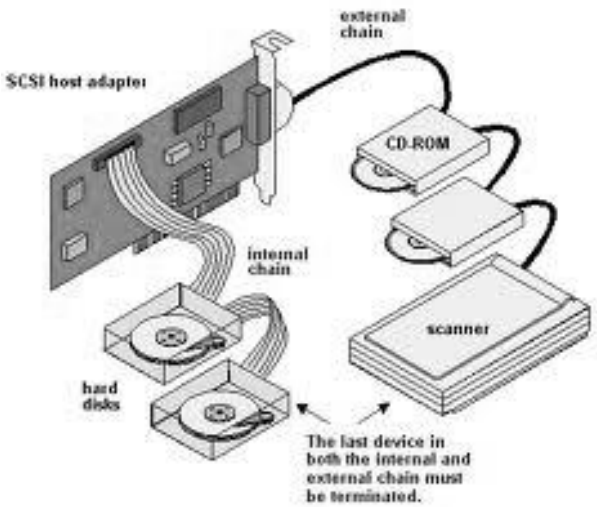


Figura 56

USB

El Universal Serial Bus (*Bus Universal* en serie USB) (Ver Figura 57) es un estándar industrial desarrollado en los años 90 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos. La iniciativa del desarrollo partió de Intel que creó el USB Implementers Forum junto con IBM, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC. Actualmente, agrupa a más de 685 compañías.

USB fue diseñado para estandarizar la conexión de periféricos, como mouse, teclados, joysticks, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, sistemas de adquisición de datos, módems, tarjetas de red, tarjetas de sonido y discos duros externos. Su éxito ha sido total, ya que ha desplazado a conectores, como el puerto serie, puerto paralelo, puerto de juegos, etcétera.

En muchos casos, el bus USB se implementa con una placa PCI como Interfaz.

Figura 57

Capítulo IV
Discos HDD, unidad sólida SSD, cintas magnéticas, discos ópticos y tarjetas de memoria

1-Disco duro o HDD (Hard Drive Disk)

En informática, un disco duro o disco rígido (en inglés, *Hard Disk Drive*, HDD) es un dispositivo de almacenamiento de datos no volátil que emplea un sistema de grabación magnética para almacenar datos digitales. Se compone de uno o más platos o discos rígidos, unidos por un mismo eje, que gira a gran velocidad dentro de una caja metálica sellada. Sobre cada plato, y en cada una de sus caras, se sitúa un cabezal de lectura-escritura que flota sobre una delgada lámina de aire generada por la rotación de los discos (Ver Figura 58.)

Figura 58

Estos discos están recubiertos por una material magnetizable. El cabezal (dispositivo de lectura y escritura) está formado por un conjunto de brazos paralelos a los platos, alineados verticalmente y que también se desplazan de forma simultánea, en cuya punta están las cabezas de lectura-escritura. Por norma general, hay una cabeza de lectura-escritura para cada superficie de cada plato. Los cabezales pueden moverse hacia el interior o el exterior de los platos, lo cual (combinado con la rotación de los mismos) permite que los cabezales puedan alcanzar cualquier posición de la superficie de los platos.

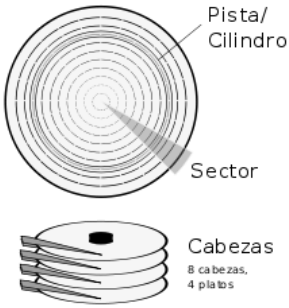
Las cabezas de lectura-escritura nunca tocan el disco, sino que pasan muy cerca (hasta a 3 nanómetros), debido a una finísima película de aire que se forma entre estas y los platos cuando estos giran (algunos discos incluyen un sistema que impide

que los cabezales pasen por encima de los platos hasta que alcancen una velocidad de giro que garantice la formación de esta película). Si alguna de las cabezas llega a tocar una superficie de un plato, causaría muchos daños en él, rayándolo gravemente, debido a lo rápido que giran los platos (uno de 7200 revoluciones por minuto se mueve a 129 km/h en el borde de un disco de 3,5 pulgadas).

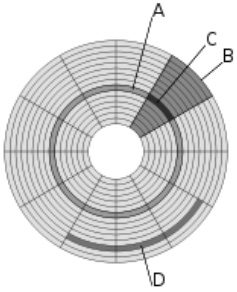
Los discos rígidos se subdividen en: plato, cara, pista, cilindro, cabeza, sector y clúster.

- Plato: cada uno de los discos que hay dentro del disco duro.
- Cara: cada uno de los dos lados de un plato.
- Pistas: una circunferencia dentro de una cara; la pista 0 está en el borde exterior. Las pistas de cada plato son concéntricas. Las pistas exteriores contienen más bits que las pistas interiores
- Cilindro: conjunto de varias pistas; son todas las circunferencias que están alineadas verticalmente (una de cada cara).
- Cabeza: número de cabezales.
- Sector : cada una de las divisiones de una pista. El tamaño del sector no es fijo, el estándar actual es de 512 bytes.
- Un clúster: un conjunto contiguo de sectores que componen la unidad más pequeña de almacenamiento de un disco. Los archivos se almacenan en uno o varios clústeres, dependiendo de su tamaño de unidad de asignación. Sin embargo, si el archivo es más pequeño que un clúster, este lo ocupa por completo, no se libera parte del clúster.

Ejemplos de división de discos rígidos:



Pista (A), Sector (B), Sector de una pista (C), Clúster (D).



Principio de funcionamiento de la lectura y la grabación

El modo de funcionamiento de un disco rígido actual guarda similitudes aún con el telegráfono de Poulsen del siglo XIX. Hay tres principios fundamentales de física básica que entran en juego.

1. Una corriente eléctrica produce un campo magnético (ley de Ampère).
2. Las variaciones del campo magnético inducen una tensión en una bobina (ley de Faraday-Lenz).
3. Los dominios en un material magnético tienden a orientarse en la dirección del campo aplicado (interacción Zeeman).

En el sistema de Poulsen, el cabezal es un electroimán (una bobina enrollada sobre un material magnético blando). Este material sirve tanto para escribir (usando las propiedades 1 y 3) como para leer la información magnética (usando la 2). El cabezal convierte la señal eléctrica en un campo magnético variable que magnetiza al medio de una manera proporcional a la intensidad de la señal. Para que la información guardada dure en el tiempo, el medio debe ser un material magnéticamente duro, o sea, que conserve su magnetización en presencia de campos externos. Si se pasa el cabezal sobre una zona determinada, los cambios de magnetización en el medio producen variaciones en el campo magnético superficial que a su vez inducen una señal eléctrica en el electroimán.

Es importante entender que la característica de la información almacenada en un disco rígido es que perdura en el tiempo a pesar de que el disco no se encuentre conectado a ninguna alimentación eléctrica. Pero, fundamentalmente, el disco duro tiene la gran virtud de que el acceso a la información es directa, es decir, que las cabezas lectoras grabadoras pueden dirigirse directamente a la pista y sector

deseado, sin tener que seguir ninguna secuencia. Por esto se dice que es un soporte de almacenamiento de acceso directo. Esta virtud hace imprescindible a los discos duros como almacenamiento para los sistemas que requieren obtener información inmediata (on line).

Actualmente, se están fabricando discos con una capacidad de 4 TB. Pero mediante la utilización de circuitos electrónicos especiales y software, se pueden integrar estos discos en lo que se denomina clúster (no confundir con la subdivisión de un disco rígido) y llega a un equivalente a 120 PB. El hardware y el software permiten integrar todos los discos y que se utilicen como si fuesen una sola unidad.

Formato

Para que los discos puedan ser usados por la computadora, se requiere que estos sean formateados a bajo nivel. El propósito del formateo de bajo nivel es dividir la superficie del disco en elementos básicos: pistas, sectores y cilindros. Esta tarea se realiza mediante la polarización de áreas del disco, utilizando los cabezales de escritura. Las pistas se numeran a partir del 0, y luego los cabezales polarizan la superficie de los discos en forma concéntrica. Cuando el cabezal pasa de una pista a la siguiente, deja un espacio. Cada pista se organiza a sí misma en sectores (con una numeración que comienza desde el 1 y se separa por espacios). Cada uno de estos sectores comienza con un área reservada para la información del sistema denominada prefijo y termina con un área denominada sufijo. El propósito del formateo de bajo nivel es, por lo tanto, el de preparar la superficie del disco para recibir datos y no depender del sistema operativo.

Durante el formateo, se lleva a cabo pruebas de control (algoritmos que permiten comprobar la validez de sectores mediante las sumas de control) y cada vez que a un sector se lo considera defectuoso, se escribe en el prefijo la suma de control (inválida). A partir de ese momento, no puede ser utilizado y se lo marca como defectuoso.

2-Unidades de estado sólido o SSD (Solid State Drive)

Las unidades de estado sólido o SSD (Solid State Drive) son una alternativa a los discos duros. La gran diferencia es que mientras los discos duros utilizan componentes mecánicos que se mueven, las SSD almacenan los archivos en microchips con memorias flash interconectadas entre sí (ver punto 6).

3-Comparación entre HDD y SDD

PRINCIPALES VENTAJAS	SSD	HDD
CAPACIDAD	En general entre 256 GB y 4 TB	En general entre 1 y 10 TB
CONSUMO	Menor consumo	Mayor consumo
COSTE	Bastante más caros	Mucho más económicos
RUIDO	Más silencioso por no tener partes móviles	Algo más ruidoso por tener partes móviles
VIBRACIONES	No vibra por no tener partes móviles	El giro de sus discos puede provocar leves vibráciones
FRAGMENTACIÓN	No tiene	Puede darse
DURABILIDAD	Sus celdas pueden reescribirse un número limitado de veces	Con partes mecánicas que pueden dañarse con movimientos
TIEMPO DE ARRANQUE DE SO	7 segundos	16 segundos
TRANSFERENCIA DE DATOS	En general, entre 200 y 550 MB/s	En general entre 50 y 150 MB/s
AFECTADO POR EL MAGNETISMO	No	El magnetismo puede eliminar datos

4-Cinta magnética

La cinta magnética es un tipo de medio o soporte de almacenamiento de datos que se graba en pistas sobre una banda plástica con un material magnetizado, generalmente óxido de hierro o algún cromato. El tipo de información que se puede almacenar en las cintas magnéticas es variado, como video, audio y datos.

Hay diferentes tipos de cintas, tanto en sus medidas físicas como en su constitución química, así como diferentes formatos de grabación, especializados en el tipo de información que se quiere grabar.

Los dispositivos informáticos de almacenamiento masivo de datos de cinta magnética son utilizados principalmente para respaldo de archivos y para el proceso de información de tipo secuencial. Para poder llegar a cualquier bit de información de una cinta magnética, es necesario pasar por todos los bits que los anteceden en la secuencia de grabación. Esta característica hace que estos soportes no sean adecuados para procesos que requieren información on line.

Actualmente, todas las cintas se encierran dentro de cajas para su mayor protección y facilidad de uso. Existe una gran variedad de cintas, en tamaños, capacidades y formatos. Solo se nombrará al más destacado de los formatos en la actualidad que son los *Linear Tape-Open* (LTO). El LTO es una cinta magnética de almacenamiento de datos, desarrollada a finales de 1990. Originalmente fue creado por Hewlett-Packard, IBM y Seagate bajo el Consorcio LTO, como alternativa a los formatos de cinta magnética patentada que estaban disponibles en ese momento (Ver Figura 59).

Figura 59

Las versiones de los LTO que han salido y que se planean liberar en los próximos años son:

	Generaciones							
	LTO-1	**LTO-2**	**LTO-3**	**LTO-4**	**LTO-5**	**LTO-6**	**LTO-7**	**LTO-8**
Release Date	2000	2003	2005	2007	2010			
Native Data Capacity	100 GB	200 GB	400 GB	800 GB	1.5 TB	3.2 TB	6.4 TB	12.8 TB

Estos cartuchos se integran en equipos que pueden administrar miles de LTO, por ejemplo, el equipo IBM System Storage TS3500 Tape Library (Ver Figura 60) que puede soportar una capacidad máxima de almacenamiento de 2.7 EB.

Figura 60

5-Discos ópticos

Actualmente, en el mercado existen tres categorías de discos ópticos: los CD, los DVD y los BLU RAY. Se explicará con más detenimiento los CD, ya que los DVD y los

BLU RAY son de similares características, con la diferencia de que pueden almacenar más información en función de algunas mejoras técnicas.

5.1-El CD

El CD o disco compacto (*Compact Disc*) es un soporte digital óptico utilizado para almacenar cualquier tipo de información: audio, imágenes, video, documentos y otros datos. Existen tres tipos de CD: los CD-ROM, los CD-R y los CD-RW

5.1.1-El CD-ROM

El CD-ROM (acrónimo de *Compact Disc Read Only Memory*) es un soporte digital óptico solo de lectura, utilizado para almacenar cualquier tipo de datos. Los CD tienen un diámetro de 12 cm y pueden almacenar hasta 700 MB de datos. La información se almacena en bits que son como baches o depresiones sobre una superficie metálica. Las depresiones representan ceros y los realces representan unos (Ver Figura 61).

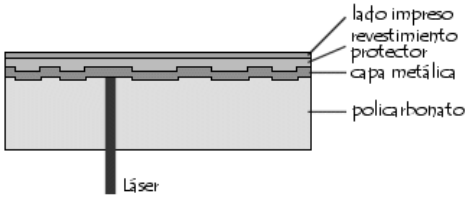


Figura 61

Los CD-ROM se fabrican con la información de la siguiente forma: 1) primero se realiza un CD metálico con los baches revertidos (al que se denomina *stamper*), que servirá como molde para realizar la totalidad de CD-ROM deseados; 2) se realiza una inyección de policarbonato dentro del molde (*stamper*); 3) se aplica la capa metálica al sustrato de policarbonato; 4) se retira el *stamper* y se procede a cubrir con una capa protectora; 5) por último, si se desea, se le imprime la etiqueta. Luego se realizan los puntos del 2 al 5 tantas veces como CD-ROM se deseen.

La lectura del CD-ROM se realiza con un cabezal en el que hay un emisor de rayos láser, que dispara un haz de luz hacia la superficie del disco y que tiene también un fotorreceptor (fotodiodo), que recibe el haz de luz que rebota en la superficie del disco. De esta manera, cuando el láser atraviesa el sustrato de policarbonato, la luz se refleja en la superficie reflectante. Sin embargo, lo que permite que se codifique la información es el acercamiento del láser a un bache. Lo que sucede es que la lectura de la refracción de la luz es diferente cuando el rayo apunta un bache o a un realce.

Un fotodiodo es una unión PN. Los diodos tienen un sentido normal de circulación de corriente, que se llama polarización directa. En ese sentido, el diodo deja pasar la corriente eléctrica y prácticamente no lo permite en el inverso. En el fotodiodo, la corriente (que varía con los cambios de la luz) es la que circula en sentido inverso al permitido por la juntura del diodo. Es decir, para su funcionamiento el fotodiodo es polarizado de manera inversa.

5.1.2-El CD-R

Los CD-R son soportes de una grabación e infinitas lecturas. Está fabricado con una capa de policarbonato a la que se le agrega una capa de un pigmento orgánico y sobre esta una capa metálica refractaria. El sustrato orgánico posee un tinte que puede ser quemado por un láser de alta potencia (10 veces más potente que el que se usa para leerlos). La capa con el tinte es la encargada de absorber o permitir el reflejo del haz de luz emitido por el láser contra la capa metálica. Una vez que el sustrato orgánico fue quemado, no puede volver a su estado anterior, por este motivo es que el CD-R solo puede grabarse una vez (Ver. Figura 62).

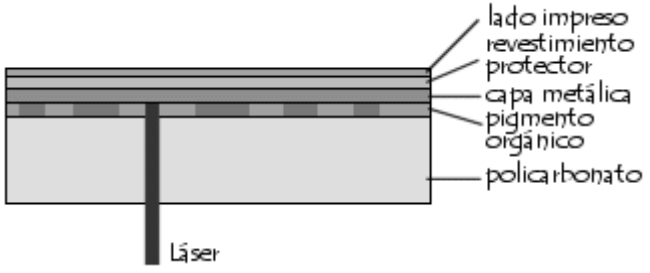


Figura 62

5.1.3-CD-RW

Un disco compacto regrabable, conocido popularmente como CD-RW, es una evolución sobre el CD-R. Este tipo de CD puede ser grabado múltiples veces, ya que permite que los datos almacenados sean borrados. Fue desarrollado conjuntamente en 1996 por las empresas Sony y Philips, y comenzó a comercializarse en 1997. En el disco CD-RW, la capa que contiene la información está formada por una aleación cristalina de plata, indio, antimonio y telurio que presenta una interesante cualidad: si se calienta hasta cierta temperatura, cuando se enfría, deviene cristalino; pero si al calentarse se alcanza una temperatura aún más elevada, cuando se enfría, queda con estructura amorfa. La superficie cristalina permite que la luz se refleje bien en la zona reflectante, mientras que las zonas con estructura amorfa absorben la luz. Por ello el CD-RW utiliza tres tipos de luz:

Láser de escritura: Se usa para escribir. Calienta pequeñas zonas de la superficie para que el material se torne amorfo.

Láser de borrado: Se usa para borrar. Tiene una intensidad menor que el de escritura, con lo que se consigue el estado cristalino.

Láser de lectura: Se usa para leer. Tiene menor intensidad que el de borrado. Se refleja en zonas cristalinas y se dispersa en las amorfas.

5.2-DVD

El DVD es un dispositivo de almacenamiento óptico cuyo estándar surgió en 1995. Sus siglas corresponden con *Digital Versatile Disc*. Existen distintos tipos de DVD: DVD-ROM (dispositivo de lectura únicamente), DVD-R y DVD+R (solo pueden escribirse una vez), DVD-RW y DVD+RW (permiten grabar y borrar las veces que se quiera). También, difieren en la capacidad de almacenamiento de cada uno de los tipos.

Para la estandarización del formato, se creó en 1995 el Consorcio del DVD (*DVD Consortium*), que en 1997 pasó a llamarse Foro DVD (*DVD Forum*) con los siguientes miembros: Hitachi, Matsushita Electric Industrial Co. Ltd., Mitsubishi Electric Corporation, Pioneer Electronic Corporation, Royal Philips Electronics N.V., Sony Corporation, Thomson, Time Warner Inc., Toshiba Corporation y Victor Company of Japan Ltd. (JVC).

El DVD es una «alternativa» al disco compacto (CD), que posee seis veces más espacio de almacenamiento (para el tipo de DVD de menor capacidad: de capa

simple y una cara). El formato DVD se diseñó para proporcionar un medio de almacenamiento universal, mientras que el CD, originalmente, se diseñó exclusivamente como un medio de audio.

5.2.1-DVD-ROM

Los DVD-ROM, que al igual que los CD-ROM ya se entregan pregrabados, son solo de lectura y también, al igual que los CD-ROM, su fabricación se realiza por intermedio de un *stamper* que permite realizar el copiado en grandes cantidades.

Existen distintos formatos de DVD-ROM:

Tipo de disco	Características	Capacidad de almacenamiento
DVD-5	una cara, capa simple	4,7 GB
DVD-9	una cara, doble capa	8,5 GB
DVD-10	dos caras, capa simple	9,4 GB
DVD-14	capa doble en una cara, capa simple en la otra	13,3 GB
DVD-17	dos caras, doble capa en cada cara	17,1 GB

Los DVD-ROM existen tanto en versiones de «capa simple» como de «doble capa». Los discos de doble capa están compuestos de una capa transparente semirreflectante de color dorado y una capa opaca reflectante de color plateado, ambas separadas por una capa de enlace. Para poder leer estas dos capas, el láser debe variar su intensidad.

- Con baja intensidad, el rayo se refleja sobre la capa dorada superior.
- Con una intensidad mayor, el rayo atraviesa la primera capa y se refleja sobre la capa plateada inferior.

Por otro lado, existen versiones de DVD-ROM tanto de una cara como de doble cara, como los discos de vinilo. En el segundo caso, la información se almacena en ambas caras del disco (Ver Figura 63).

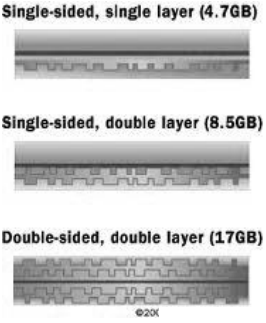


Figura 63

5.2.2-DVD-R

El DVD-R solo puede grabarse una vez. Existen en formatos 5, 9, 10 y 17. La tecnología de grabación y lectura es muy similar a la del CD-R. Los láseres de DVD requieren mucha potencia para alcanzar la segunda capa y poder quemar el sustrato orgánico. Las capas de un DVD-R con doble capa de grabación son las siguientes:

La capa para la impresión
Un recubrimiento de policarbonato
Capa metálica refractaria
Sustrato orgánico (primer nivel de datos)
Una capa de policarbonato para separación y adhesión.
Una capa semirrefractaria para que permita pasar el láser que busca la primera capa refractaria.
Sustrato orgánico (segundo nivel de datos)
Una capa de policarbonato.

5.2.3-DVD+R

El DVD+R solo puede grabarse una vez. Existen en formatos 5, 9, 10 y 18. La tecnología de grabación y lectura es muy similar a la del CD-R. Tanto este formato como el DVD+RW están patrocinados por Sony y Philips dentro de la DVD+RW Alliance, que también incluye a Dell, Hewlett-Packard, Mitsubishi/Verbatim, Ricoh, Thomson y Yamaha.

Las capas del DVD+R son las mismas que las del -R. Las diferencias entre el formato -R y el +R son de funcionalidades, cada una con sus ventajas y desventajas; pero, en general, se pude afirmar que el DVD-R es preferible para crear DVD de video, mientras que el DVD+R resulta más apropiado para crear discos de datos. La mayoría de las grabadoras de DVD son compatibles con ambos formatos.

5.2.4-DVD-RW

El DVD-RW se puede grabar y borrar en múltiples oportunidades. La tecnología de grabación y lectura es muy similar a la del CD-RW.

5.2.5-DVD+RW

El DVD+RW se puede grabar y borrar en múltiples oportunidades. La tecnología de grabación y lectura es muy similar a la del CD-RW. Las diferencias entre el DVD-RW y +RW son funcionales y tienen las mimas ventajas y desventajas que el DVD-R y y DVD+R. La recomendación, nuevamente, es que el DVD-RW es preferible para crear DVD de video, mientras que el DVD+RW resulta más apropiado para crear discos de datos. La mayoría de las grabadoras de DVD son compatibles con ambos formatos.

5.3-BLU-RAY

Blu-ray disc, también conocido como Blu-ray o BD, es un formato de disco óptico de nueva generación de 12 cm de diámetro (igual que el CD y el DVD) para video de gran definición y almacenamiento de datos de alta densidad. Su capacidad de almacenamiento llega a 25 GB por capa.

El disco Blu-ray hace uso de un rayo láser de color azul con una longitud de onda de 405 nanómetros, a diferencia del láser rojo utilizado en lectores de DVD, que tiene una longitud de onda de 650 nanómetros. Esto, junto con otros avances tecnológicos, permite almacenar sustancialmente más información que el DVD en un disco de las mismas dimensiones y aspecto externo. Fue desarrollado en conjunto por un grupo de compañías tecnológicas llamado Blu-Ray Disc Association (BDA), liderado por Sony y Philips.

El tamaño del punto mínimo en el que un láser puede ser enfocado está limitado por la difracción y depende de la longitud de onda del haz de luz y de la apertura numérica de la lente utilizada para enfocarlo. En el caso del láser azul-violeta utilizado en los discos Blu-ray, la longitud de onda es menor con respecto a tecnologías anteriores, aumentando por tanto la apertura numérica (0,85 comparado con 0,6 para DVD). Con ello, y gracias a un sistema de lentes duales y a una cubierta protectora más delgada, el rayo láser puede enfocar de forma mucho más precisa en la superficie del disco. Dicho de otra forma, los puntos de información legibles en el disco son mucho más pequeños y, por lo tanto, el mismo espacio puede contener mucha más información. Por último, además de las mejoras en la tecnología óptica, estos discos incorporan un sistema mejorado de codificación de datos que permite empaquetar aún más información.

El DVD tenía dos problemas que se intentaron resolver con la tecnología Blu-Ray, por ello la estructura es distinta. En primer lugar, para la lectura en el DVD, el láser debe atravesar la capa de policarbonato de 0,6 mm en la que el láser se puede difractar en dos haces de luz. Si esta difracción es alta, por ejemplo, si estuviera rayado, impide la lectura del disco. Pero dicho disco, al tener una capa de solo 0,1 mm evita este

problema, ya que tiene menos recorrido hasta la capa de datos; además, esta capa es resistente a rayaduras. En segundo lugar, si el disco estuviera inclinado, en el caso del DVD, por igual motivo que el anterior problema, la distorsión del rayo láser haría que leyese en una posición equivocada, dando lugar a errores. Gracias a la cercanía de la lente y la rápida convergencia del láser, la distorsión es inferior y se pueden evitar posibles errores de lectura.

Existen blu-ray de una capa y de dos capas y de los siguientes tipos:

- BD-ROM: formato solo de lectura para la reproducción y distribución de películas de alta definición, videojuegos y software. Su forma de lectura es similar a la del CD-ROM.
- BD-R: formato para grabación por única vez que puede ser utilizado para grabar películas de alta definición o datos en una computadora. Su forma de lectura y grabación son similares al CD-R.
- BD-RE: formato regrabable para almacenar películas de alta definición o datos de computadora. Su forma de lectura y grabación son similares a las del CD-RW.

6-Tarjeta de memoria o memoria flash

Una tarjeta de memoria o tarjeta de memoria flash es un dispositivo de almacenamiento que conserva la información que ha sido almacenada de forma correcta aun con la pérdida de energía, es decir, es una memoria no volátil.

Hay una gran diversidad de tarjetas de memoria. El término Memoria Flash fue acuñado por Toshiba, por su capacidad para borrarse «en un flash» (instante). Se borran en bloques fijos, en lugar de bytes solos. Los tamaños de los bloques por lo general van de 512 bytes hasta 256 KB. Los chips flash son poco costosos y proporcionan grandes densidades de bits.

La PC Card (PCMCIA) se encontraba entre los primeros formatos comerciales de tarjetas de memoria que salen en la década de 1990, pero ahora se utiliza principalmente en aplicaciones industriales y para conectar dispositivos de Entrada-Salida. También, en los años 90, una serie de formatos de tarjetas de memoria más pequeña que la PC Card salieron a la luz, incluyendo CompactFlash, SmartMedia, Secure Digital, MiniSD, MicroSD y similares. El deseo de pequeñas tarjetas en teléfonos móviles, PDAs y cámaras digitales compactas produjo una tendencia que dejó la anterior generación de tarjetas demasiado grandes. En las cámaras digitales

SmartMedia y CompactFlash, había tenido mucho éxito. En 2001, SM había capturado el 50% del mercado de cámaras digitales y CF tenía un dominio absoluto sobre las cámaras digitales profesionales. En 2005, sin embargo, Secure Digital/Multi Media Card había ocupado el puesto de SmartMedia, aunque no al mismo nivel y con una fuerte competencia procedente de las variantes de Memory Stick, xD-Picture Card y CompactFlash. A partir del año 2010, microSD pasa a dominar el mercado de smartphones, tabletas y cámaras fotográficas.

6.1-Las tarjetas microSD

Las tarjetas microSD o Transflash corresponden a un formato de tarjeta de memoria flash más pequeña que la MiniSD, desarrollada por SanDisk y adoptada por la Asociación de Tarjetas SD bajo el nombre de «microSD» en julio de 2005. Mide tan solo 15 × 11 × 1 mm, lo cual le da un área de 165 mm^2. Esto es tres veces y media más pequeña que la miniSD, que era hasta la aparición de las microSD el formato más pequeño de tarjetas SD, y es alrededor de un décimo del volumen de una tarjeta SD. Sus tasas de transferencia no son muy altas, sin embargo, empresas como SanDisk han trabajado en ello, llegando a versiones que soportan velocidades de lectura de hasta 10 Mb/s. Actualmente, ya existen tarjetas microSD fabricadas por Panasonic que alcanzan los 90 Mb/s de lectura y los 80 Mb/s de escritura, pero tienen unos precios todavía muy elevados. Ahora existe una tarjeta más avanzada que la microSD que se denomina microSDHC. Los equipos preparados para microSDHC soportan microSD, mas no ocurre lo inverso.

Las capacidades que se comercializan son las siguientes:

MicroSD:

- 1 GB Micro SD de 1 GB de capacidad.
- 2 GB
- 4 GB
- 8 GB
- 16 GB
- 32 GB
- 64 GB

MicroSDHC

- 2GB
- 4 GB

- 8 GB
- 16 GB
- 32 GB
- 64 GB
- 128 GB

Funcionamiento del MicroSD

Las tarjetas MicrosSD no utilizan un condensador para almacenar el bit como lo hacen las memorias SDRAM, sino que utilizan un transistor denominado FGMOS (Ver Figura 64), que son muy similares a los NMOS y que se explicará muy brevemente, ya que hacerlo con rigor científico sería en extremo complejo.

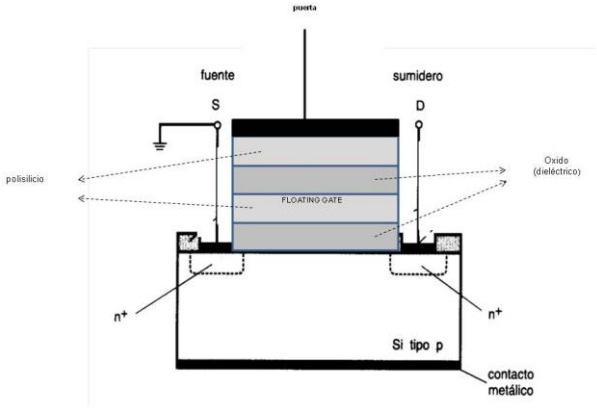


Figura 64

La diferencia entre el NMOS y el FGMOS es la puerta o *gate* que está formada por varias capas. Empezando por la parte inferior, se coloca una capa de óxido como material dieléctrico; luego, sobre el óxido, una capa de polisilicio que es la capa denominada *floting gate.* Esta capa cargada de electrones representa un bit en un 0 lógico, por el contrario, si esta capa está pobre de electrones, significará un 1 lógico. Después, sobre la capa de *floting gate*, se coloca otra capa de óxido como dieléctrico y por sobre esta otra capa más de polisilicio y recién a esta capa se le conecta el metal de la puerta del transistor.

El proceso de grabación

Para cargar el *floating gate*, se debe ingresar por la puerta del transistor 20 volts, tal como lo muestra la figura 65. De esta forma, la capa de *floating gate* se cargará de electrones. Siempre con fuente y sumidero desconectados. Así se crea el 0 lógico.

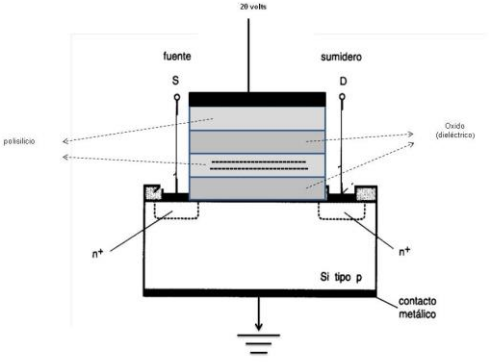


Figura 65

Invirtiendo la polaridad, se descarga la capa de *floating gate* y se forma el túnel entre los cristales semiconductores n para que pueda fluir la electricidad entre ambos (Ver Figura 66). Así se crea el 1 lógico.

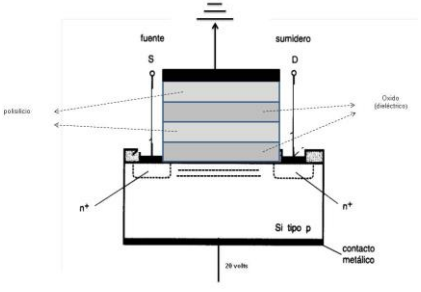


Figura 66

El proceso de lectura

Se ingresan 3 volts por la puerta del transistor y una pequeña corriente por la fuente. Si se detecta corriente en el sumidero, es porque el *floating gate* está descargado y el túnel entre las placas n está formado. Esto significa que el transistor está en 1 lógico. Si, por el contrario, no se detecta corriente en el sumidero, esto significa que el *floating gate* está cargado, por lo tanto, el transistor está en un 0 lógico.

Capítulo V

Códigos de barras, códigos de barra bidimensionales, código QR, MICR, escáner, reconocimiento de voz, RFID, cámaras digitales, teclado, mouse y touchpad

1-Códigos de barra

Un código de barras es una representación gráfica de un dato que utiliza barras como elemento básico de representación. Aunque se parecen, no todos los códigos de barras son iguales. Existen diferentes maneras de representar el mismo dato, utilizando distintos patrones de barras. Lo que diferencia unas de otras es básicamente el ancho de las barras, aunque existen otros criterios. Por ejemplo, los dos tipos de codificación: EAN-13 y Código 39

EAN-13

Los productos de consumo masivo utilizan la codificación EAN-13, que constan, precisamente, de 13 dígitos. Los tres primeros, de izquierda a derecha, corresponden al país de origen. Los nueve dígitos que siguen identifican al fabricante y al producto. El último es un dígito de verificación. El dígito verificador es un dígito que corresponde a una relación matemática de los 12 dígitos anteriores. Este se utiliza para verificar que los 12 dígitos precedentes hayan sido bien leídos, porque de no ser así, el cálculo matemático del dígito verificador dará un número diferente. La técnica de dígito verificador es muy utilizada para este tipo de lecturas e incluso para verificar el ingreso de datos por teclado.

La codificación de barra está basada en el sistema de numeración binario, y los códigos de barra son una muestra de la infinidad de aplicaciones que pueden desarrollarse utilizando solamente ceros y unos. Se puede observar cómo se interpretan las barras.

Figura 67

En la figura 67, se ve una representación de un 4 en el binario de EAN-13. La tabla de representación binaria del código EAN-13 es lo bastante compleja como para dedicarle más explicación. Es suficiente con entender que de la secuencia de barras y espacios es posible obtener un código binario y que dicho código binario tiene su representación decimal.

Como se puede observar, es común agregar bajo el código de barra la representación en decimal. En general, en cualquier aplicación, es buena idea incluirlos, ya que en ocasiones hay que teclear el código porque el lector óptico no lo logra leer. Esto se debe principalmente a manchas, pliegues, tinta borrosa, etcétera.

Código 39

Código 39 (Ver Figura 68) es un código de barras capaz de representar letras mayúsculas, números y algunos caracteres especiales, como el espacio. Posiblemente la mayor desventaja del código es su baja densidad de impresión.

Código 39 (o Código 3 de 9) es el código de barras más habitual. Se utiliza muy a menudo porque:

- Admite tanto texto como números (A-Z, 0-9, +, -, , y <espacio>).
- Puede ser leído por casi cualquier lector de código de barras con su configuración predeterminada.
- Es, dentro de los códigos de barras modernos, uno de los más antiguos.

Figura 68

Concepto de código público o privado

Se denominan códigos de uso privado a aquellos que deben pagar regalías (o derechos), por su uso. Esta regalía debe pagarse se lea o no el código, es decir, por el derecho a imprimirlo.

Solamente existen dos simbologías privadas: el UPC (*Universal Product Code*, utilizado en los Estados Unidos y Canadá) y el EAN (*European Article Number*, utilizado en el resto del mundo). En la Argentina, se usa el EAN con sus variantes EAN-8 y EAN-13, y los productos fabricados en nuestro país son fácilmente distinguibles porque comienzan con el número 77.

El código EAN está regulado en la Argentina por la organización GS1 Argentina que depende de la organización mundial GS1.

Los códigos públicos, como su nombre lo indica, se pueden utilizar libremente para cualquier propósito sin pagar ningún tipo de regalía. Las simbologías más usadas en este caso son el código 39 y el código 128.

Los lectores

Hay dos tipos de lectores: láser y CCD (*Charged Coupled Device*, Dispositivo de Carga Acoplada).

La tecnología CCD se basa en tomar una fotografía de la imagen. Un conjunto de leds iluminan el código y la luz reflejada «carga» eléctricamente a los diodos que funcionan a modo de censores fotoeléctricos. De ahí el nombre de carga acoplada: una parte emite luz y otra se carga eléctricamente.

Esta tecnología tiene la ventaja de que, al no existir partes móviles, requiere menor mantenimiento, aparte de tender a ser el más económico. Sin embargo, una relativa desventaja es la distancia a la que se puede leer el código: en el mejor de los casos es a 20 cm.

Otra desventaja es la velocidad a la que puede trabajar el lector. Solo se puede leer de frente y tiene un estrecho foco angular.

El láser se basa en un haz que «barre» el código de ida y vuelta una y otra vez hasta que lo lee por completo. La ventaja en el uso de esta tecnología es principalmente la distancia a la que se puede leer, que en algunos casos es de hasta de 10 metros.

El barrido se logra haciendo que el haz láser incida sobre un espejito, el cual está montado sobre un motor que se mueve de ida y vuelta muy rápidamente. La refracción del láser es tomada por un fotosensor que transforma la luz en energía eléctrica. Este movimiento es el que hace que aparezca la conocida línea roja. Al tener partes móviles, un lector láser es más caro y requiere de mantenimiento, pero es la única tecnología con la que se pueden lograr grandes distancias de lectura.

Lectores multilínea o *multiraster*

Estos lectores no requieren acomodar el código en un sentido para que lo identifique el lector. Esto se debe a que cuentan con lectores multilínea. La tecnología para hacer este tipo de lectores es láser, aquí se requieren más espejos y motores para lograr que se vea no una línea, sino una especie de rejilla de líneas rojas.

Códigos de barras infrarrojos

Son códigos de barra que están cubiertos por una capa opaca que impide su fotocopiado o lectura con un lector común. Se requiere un lector infrarrojo. Estos códigos se utilizan en aplicaciones donde se necesita mucha seguridad y confidencialidad.

2-Códigos de barras bidimensionales

Existen códigos que guardan la información utilizando la dimensión de las barras en ancho como en alto, o sea, en dos dimensiones. El código «clásico» de dos dimensiones es el PDF 417 (*Portable Data File 417*) (Ver Figura 69).

Figura 69

¿Qué ventajas tienen las simbologías de dos dimensiones sobre los de una? Básicamente, es la cantidad de información que se puede codificar con ellas. Es un código de interpretación binaria compleja.

3-Código QR

Un código QR (*Quick Response Barcode*, Código de barras de respuesta rápida) es un sistema para almacenar información en una matriz de puntos o un código de barras bidimensional creado por la compañía japonesa Denso Wave, subsidiaria de Toyota, en 1994. Se caracteriza por los tres cuadrados que se encuentran en las esquinas y que permiten detectar la posición del código al lector. Los códigos QR son muy comunes en Japón y de hecho son el código bidimensional más popular en ese país (Ver Figura 70).

Figura 70

Aunque inicialmente su uso se restringió a la fabricación de vehículos, hoy los códigos QR se usan para administración de inventarios en una gran variedad de industrias. Recientemente, la inclusión de software que lee códigos QR en teléfonos móviles ha permitido nuevos usos orientados al consumidor, que se manifiestan en comodidades, como el dejar de tener que introducir datos de forma manual en los teléfonos. Las direcciones y los URLs se están volviendo cada vez más comunes en revistas y anuncios. El agregado de códigos QR en tarjetas de presentación también se está haciendo común, simplificando en gran medida la tarea de introducir detalles individuales de un nuevo cliente en la agenda de un teléfono móvil.

Un detalle muy importante sobre el código QR es que su código es abierto y que sus derechos de patente (propiedad de Denso Wave) no son ejercidos.

El código QR soporta hasta 7089 valores numéricos y hasta 4296 alfanuméricos.

4-MICR

Esta lectura se basa en el reconocimiento de la identificación de característica física de los caracteres, valiéndose de la utilización de tinta magnetizada. Algunos ejemplos son los cheques y otros instrumentos comerciales, como papel moneda o sistemas mecanizados de correspondencia. Los cheques por lo general presentan estos caracteres o números impresos en su parte inferior para su procesamiento automático mediante medios magnéticos.

En la Argentina y fundamentalmente en Europa se emplea un código conocido por sus siglas como CMC7, donde la figura de cada número está formada por siete líneas verticales y siete espacios. Mediante esta forma de impresión, se integra un código binario de unos y ceros respectivamente; según su distribución, definen los números del 0 al 9, y adicionando algunos símbolos, se puede identificar magnéticamente el banco, sucursal, número de cuenta y número de cheque (Ver Figura 71).

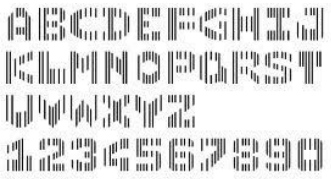


Figura 71

Para la lectura, se utiliza un lector magnético similar a los cabezales de las cintas magnéticas.

En los Estados Unidos y México, se utiliza con el mismo objetivo el código E13B, que recibe su nombre del proceso efectuado por el lector magnético, consistente en leer en sentido vertical líneas de trece milésimas de pulgada que condensan información similar a la citada para el caso europeo (Ver Figura 72).

Figura 72

5-Escáner

Un escáner es un periférico de captura utilizado para escanear documentos; es decir, convertir un documento en papel en una imagen digital.

En general, se puede decir que existen tres tipos de escáner:

- Los escáneres planos permiten escanear un documento colocándolo de cara al panel de vidrio. Este es el tipo de escáner más común.
- Los escáneres manuales son de tamaño similar. Estos deben desplazarse en forma manual (o semimanual) en el documento, por secciones sucesivas si se pretende escanearlo por completo.
- Los escáneres con alimentador de documentos hacen pasar el documento a través de una ranura iluminada para escanearlo, de manera similar a las máquinas de fax.

También existen escáneres capaces de escanear elementos específicos, como diapositivas, negativos, películas, planos de arquitectura, etcétera.

El principio de funcionamiento de un escáner es el siguiente:

- El escáner se mueve a lo largo del documento, línea por línea, emitiendo una luz potente sobre el documento.

- De la refracción de la luz, se reciben las líneas y se divide en «puntos básicos», la cantidad de puntos básicos depende de la resolución a la cual se escanea, por ejemplo 2400 dpi (que son puntos o píxeles por pulgada).
- Unas células fotoeléctricas (diodos) pasan la luz a corriente y en función de la intensidad de cada lectura determinan el color de cada píxel.
- El color de cada píxel se divide en 3 componentes (rojo, verde, azul).
- Cada componente de color se almacena en función de una tabla binaria que será de 8, 16, 32 bits. Cuantos más bits se utilizan, más cantidad de colores tiene definida la tabla de conversión y, por lo tanto, más definida en colores será la imagen. En el archivo que se crea al escanear, cada punto está representado por tres colores en la cantidad de bits que se hayan dispuesto y, además, se almacena la posición real de cada punto en la imagen escaneada. Como las imágenes ocupan una gran cantidad de información, se utilizan técnicas de compresión de información, como los archivos GIF o JPG. Por ejemplo, una imagen de 5 pulgadas x 5 pulgadas sin comprimir, escaneada a 2400 píxeles x pulgada, requiere de 25 pulgadas cuadradas, y si la imagen se almacena en 16 bits, en total requiere de 2.880.000 bits, que es igual a 2,88 MB. La cuenta es de la siguiente forma: 25 pulgadas cuadradas x 2400 pixeles x tres colores para cada píxel, x 16 bits, que es lo que requiere cada color, dan los 2,88 MB.

6-Reconocimiento de voz

El proceso de reconocimiento de voz está basado en software más que en hardware. Todo programa de reconocimiento de la voz requiere un breve período de «entrenamiento» de la computadora: una vez cargado el software y conectado el micrófono, hay que dejar que el programa guíe al usuario, pidiéndole, entre otras cosas, leer un texto en voz alta ante el micrófono. De este modo, el programa se adapta a cada voz. Una vez finalizado el entrenamiento, se podrá dictarle a la computadora, en el tono y ritmo con que se entrenó a la computadora.

Al hablar frente al micrófono, la voz se envía a la computadora como una señal analógica creada por una pequeña bobina y un imán en el interior del micrófono. El software de reconocimiento de voz convierte la señal analógica en una digital y la procesa para ecualizar el volumen, el tono, la velocidad y otras variantes. Un modelo acústico compara las palabras ingresadas en el entrenamiento y elige las palabras que se ajustan mejor al sonido de lo que se grabó. Un modelo de lenguaje utiliza el contexto de las palabras para encontrar opciones probables.

Luego, un mecanismo de voz combina la información acústica y del lenguaje para adivinar las palabras que se pronunciaron. El programa escribe en pantalla su mejor opción. Para finalizar, existe una etapa de retroalimentación, cuando se corrige una

palabra que no se reconoció en forma adecuada, la computadora modifica los modelos acústicos y de lenguaje para lograr mayor precisión la próxima vez.

7-RFID

RFID (siglas de *Radio Frequency IDentification*, en español Identificación por radiofrecuencia) es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas Auto ID (*Automatic Identification* o Identificación Automática).

Las etiquetas RFID son unos dispositivos pequeños, similares a una pegatina, que pueden ser adheridas o incorporadas a un producto, un animal o a una persona (Ver Figura 73). Contienen antenas que les permite recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID (Ver Figura 74).

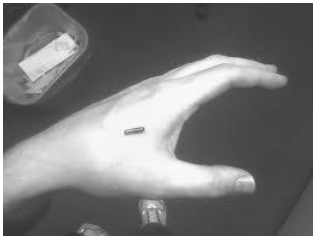


Figura 73

Figura 74

El modo de funcionamiento de los sistemas RFID es simple. La etiqueta RFID, que contiene los datos de identificación del objeto al que se encuentra adherido, genera una señal de radiofrecuencia con dichos datos. Esta señal puede ser captada por un

lector RFID, el cual se encarga de leer la información y pasarla en formato digital a la aplicación específica que utiliza RFID.

La señal que les llega de los lectores induce una corriente eléctrica pequeña y suficiente para operar el tag que está formando por circuito integrado con transistores CMOS, de forma que puede generar y transmitir una respuesta (Ver Figura 75). El transistor CMOS (*Complementary metal-oxide-semiconductor*) es una de las familias lógicas empleadas en la fabricación de circuitos integrados. Su principal característica consiste en la utilización conjunta de transistores de tipo PMOS y tipo NMOS.

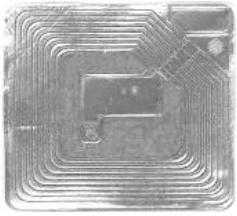


Figura 75

Un sistema RFID consta de los siguientes tres componentes:

Etiqueta RFID o transpondedor: compuesta por una antena, un transductor radio y un material encapsulado o chip. El propósito de la antena es permitirle al chip transmitir la identificación de la etiqueta. El chip posee una memoria interna con una capacidad que depende del modelo y varía de una decena a millares de bytes. Existen varios tipos de memoria:

Solo lectura: el código de identificación que contiene es único y es personalizado durante la fabricación de la etiqueta.

De lectura y escritura: la información de identificación puede ser modificada por el lector.

Lector de RFID o *transceptor*: compuesto por una antena, un transceptor y un decodificador. El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones. Cuando capta una señal de una etiqueta (la cual contiene la información de identificación de esta), extrae la información y se la pasa al subsistema de procesamiento de datos.

Subsistema de procesamiento de datos o *Middleware RFID:* proporciona los medios de proceso y almacenamiento de datos.

Las etiquetas RFID pueden ser pasivas, semipasivas (también conocidos como semiactivos o asistidos por batería) o activas:

Los tags pasivos: no requieren ninguna fuente de alimentación interna y son dispositivos puramente pasivos (solo se activan cuando un lector se encuentra cerca para suministrarles la energía necesaria). Los otros dos tipos necesitan alimentación, típicamente una pila pequeña.

La gran mayoría de las etiquetas RFID son pasivas, que son mucho más baratas de fabricar y no necesitan batería. En 2004, estas etiquetas tenían un precio desde U$S 0,40, en grandes pedidos, para etiquetas inteligentes, según el formato, y de U$S 0,95 para tags rígidos usados frecuentemente en el sector textil encapsulados en PPs o epoxi.

El mercado de RFID para la comercialización de productos será viable cuando el volumen de unidades al año supere los 10.000 millones, llevando el costo de producción a menos de U$S 0,05. La demanda actual de chips de circuitos integrados con RFID no está cerca de soportar ese costo.

Los tags pasivos suelen tener distancias de uso práctico comprendidas entre los 10 cm (ISO 14.443) y hasta unos pocos metros (EPC e ISO 18000-6), según la frecuencia de funcionamiento y el diseño y tamaño de la antena. Por su sencillez conceptual, son obtenibles por medio de un proceso de impresión de las antenas. Como no precisan de alimentación energética, el dispositivo puede resultar muy pequeño: pueden incluirse en una pegatina o insertarse bajo la piel (tags de baja frecuencia).

Los tags activos: a diferencia de los tags pasivos, los activos poseen su propia fuente autónoma de energía, que utilizan para dar corriente a sus circuitos integrados y propagar su señal al lector. Estos tags son mucho más fiables (tienen menos errores que los pasivos) debido a su capacidad de establecer sesiones con el lector. Gracias a su fuente de energía, son capaces de transmitir señales más potentes que las de los tags pasivos, lo que los hace más eficientes en entornos dificultosos para la radiofrecuencia. También son efectivos a distancias mayores y pueden generar respuestas claras a partir de recepciones débiles (lo contrario que los tags pasivos). Por el contrario, suelen ser mayores y más caros, y su vida útil es en general mucho más corta. Ver ejemplo de tag activo para reconocimiento de vehículos en la figura 76.

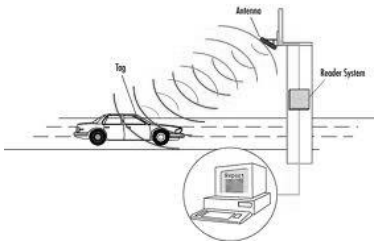


Figura 76

Muchos tags activos tienen rangos efectivos de cientos de metros y una vida útil de sus baterías de hasta diez años. Algunos de ellos integran sensores de registro de temperatura y otras variables que pueden usarse para monitorizar entornos de alimentación o productos farmacéuticos. Otros sensores asociados con RFID incluyen humedad, vibración, luz, radiación, temperatura y componentes atmosféricos, como el etileno. Se utilizan frecuencias de hasta 868 MHz (UHF) y de 2,4 GHz (banda ISM: industrial, científica y médica), la misma que para dispositivos wireless LAN 802.11b.

Tags semipasivos: se parecen a los activos en que poseen una fuente de alimentación propia, aunque en este caso se utiliza principalmente para alimentar el microchip y no para transmitir una señal. La energía contenida en la radiofrecuencia se refleja hacia el lector como en un tag pasivo. Un uso alternativo para la batería es almacenar información propagada desde el lector para emitir una respuesta en el futuro. Los tags sin batería deben responder reflejando energía de la portadora del lector al vuelo.

La batería puede permitir al circuito integrado de la etiqueta estar constantemente alimentado y eliminar la necesidad de diseñar una antena para recoger potencia de una señal entrante. Las etiquetas RFID semipasivas responden más rápidamente, por lo que son más fuertes en la ratio de lectura que las pasivas.

Este tipo de tags tienen una fiabilidad comparable a la de los tags activos, a la vez que pueden mantener el rango operativo de un tag pasivo. También suelen durar más que los tags activos.

Regulación de frecuencias: No hay ninguna corporación pública global que gobierne las frecuencias usadas para RFID. En principio, cada país puede fijar sus propias reglas. En la Argentina, las reglas las fija la CNC (Comisión Nacional de Comunicaciones).

Regulación del protocolo RFID: existe el estándar ISO que regula la normativa para la comunicación RFID.

EPC

El código electrónico de producto (código EPC, por sus siglas en inglés *Electronic Product Code*) es un número único diseñado para identificar de manera inequívoca cualquier objeto. Este código es un sistema de identificación y seguimiento de productos. El número se encuentra almacenado en un circuito integrado, denominado tag, que puede leerse mediante radiofrecuencia RFID. Puede considerarse como la evolución del código EAN (Europa) o UPC (América) y proporciona datos adicionales al clásico código de barras.

Las diferencias prácticas entre el código EAN y el código EPC se pueden resumir en:

- Ya no hay diferencias entre países o zonas de influencias; el sistema de codificación es igual para todos los países del mundo.
- Está compuesto por 24 dígitos en lugar del los 13 del código EAN.
- Los últimos 9 números hacen de numerador, de tal forma que es posible numerar más de 68 billones artículos de un mismo producto sin repetir el código.

EPCglobal es la organización que regula el sistema EPC y es una extensión de GS1. Su papel primordial es el de asesorar y homologar las aplicaciones disponibles en la industria así como las empresas reconocidas como integradoras.

8-Cámara digital

Existen dos tipos de cámara digitales como medio de input para computadoras: las que tienen un respaldo CCD y las que tienen respaldo CMOS:

Las CCD o (*Charge-Coupled Device*, en español Dispositivo de carga acoplada) es un circuito integrado que contiene un número determinado de condensadores enlazados o acoplados. Bajo el control de un circuito interno, cada condensador puede transferir su carga eléctrica a uno o a varios de los condensadores que estén a su lado en el circuito impreso. La carga de los condensadores se produce en función de la energía producida por diodos que transforman la luz obtenida por el lente en energía.

La alternativa digital a los CCD son los dispositivos CMOS utilizados en algunas cámaras digitales y en numerosas cámaras web. En la actualidad, los CCD son mucho más populares en aplicaciones profesionales y en cámaras digitales autónomas.

9-Teclado

El teclado habitual consta de 102 teclas (101 teclas en la versión americana); también hay que tener en cuenta los teclados especiales de las máquinas portátiles, de formato reducido y que exigen maniobras complicadas para simular el funcionamiento de las 102 teclas de referencia.

Debajo de las teclas, se encuentra una placa electrónica bastante compleja que consta de un microprocesador (un microcontrolador) especializado. Esta tarjeta garantiza la parte esencial del tratamiento electrónico. Esta placa electrónica contiene una matriz de contactos, que al presionar una tecla cierran el circuito. El microcontrolador detecta la presión de la tecla y genera un código («código de activación» conocido como *scan code*). Al soltarse la tecla, se genera otro código («reposo» o *break code*). La rutina de teclado del BIOS traduce el código de la tecla al código ASCII que será recibido por la CPU.

10-Mouse

El mouse es un dispositivo apuntador utilizado para facilitar el manejo de un entorno gráfico en una computadora. Existen de diferentes tecnologías y tamaños.

Están integrados a varios comandos: 1-Scroll, o botón inteligente, que consta de un rodillo externo que permite girar hacia abajo y arriba, esto permite mover fácilmente las pantallas, además de tener la función de un botón, ya que se puede presionar para mantener la pantalla en movimiento. 2-Botones de función, son botones laterales que permiten su programación para ser utilizados de múltiples maneras en las diferentes aplicaciones.

10-1-Mecánicos (Ver Figura 77)

Consta de una bola de silicona cubierta de caucho (goma) que gira en la parte inferior del ratón. Esta esfera transfiere el movimiento a un par de rodillos de plástico con perforaciones rectangulares, uno registra los movimientos horizontales y el otro los verticales, la suma de ambos da el movimiento real. Cada rueda de plástico se encuentra en medio de un LED (diodo emisor de luz) y un fotorreceptor. Cuando el LED emite luz, esta se propaga en línea recta y si no encuentra interferencia con alguna rejilla de la rueda de plástico, llega hasta el fotorreceptor, el cual, a su vez, genera un impulso eléctrico. El movimiento de cada rueda de plástico, aunado al sistema LED-fotorreceptor, genera una serie de impulsos eléctricos. Entre más rápidos sean los impulsos, mayor será la rapidez de movimiento del cursor. Por último, las señales generadas por los dos sistemas (LED-fotorreceptor) se unen en el circuito integrado que controla el ratón; posteriormente, son enviadas a la

computadora a través del cable que los une o por medio de algún sistema inalámbrico.

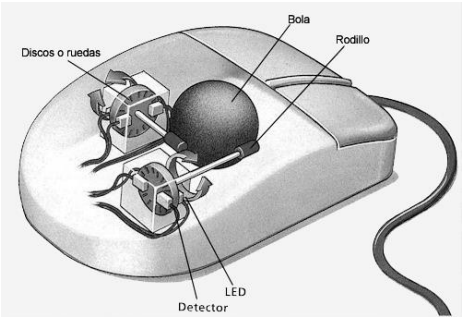


Figura 77

10-2 Ópticos (Ver figura 78)

Es una variante que carece de la bola de goma que evita el frecuente problema de la acumulación de suciedad en el eje de transmisión. Por sus características ópticas, es menos propenso a sufrir un inconveniente similar. Se considera uno de los más modernos y prácticos actualmente. Puede ofrecer un límite de 800 ppp, como cantidad de puntos distintos que puede reconocer en 2,54 cm (una pulgada); a menor cifra, peor actuará el sensor de movimientos. Los ratones ópticos cuentan con un LED que dispara un rayo de luz infrarroja sobre una superficie que lo refleja para ser capturado por un receptor del mouse, que es un chip sensible a la luz denominado CCD, parecido al que tienen las cámaras digitales, el cual envía la información a un procesador de señal, que registra el cambio de reflexión de la luz, la velocidad y la dirección, y tomando en cuenta los factores anteriores, se determina el movimiento horizontal y vertical del ratón. El CCD recibe y procesa 1500 reflejos por segundo, lo que da un seguimiento digital del movimiento y, por tanto, una precisión extraordinaria.

Figura 78

10-3 Láser

Este tipo es más sensible y preciso, lo que lo hace aconsejable especialmente para los diseñadores gráficos y los jugadores de videojuegos. También detecta el movimiento deslizándose sobre una superficie horizontal, pero el haz de luz de tecnología óptica se sustituye por un láser con resoluciones a partir de 2000 ppp, lo que se traduce en un aumento significativo de la precisión y sensibilidad.

10-4 Trackball

El concepto de trackball (Ver Figura 79) es una idea que parte de que se debe mover el puntero, no el dispositivo, por lo que se adapta para presentar una bola, de tal forma que cuando se coloque la mano encima se pueda mover mediante el dedo pulgar, sin necesidad de desplazar toda la mano como con el mouse. De esta manera, se reduce el esfuerzo y la necesidad de espacio, además de evitar el posible dolor de antebrazo por el movimiento de este. A algunas personas, sin embargo, no les termina de resultar realmente cómodo. Este tipo ha sido muy útil, por ejemplo, en la informatización de la navegación marítima.

Figura 79

11-Touchpad

Son los utilizados normalmente en los ordenadores portátiles para suplir al ratón. El touchpad (Ver Figura 80) está formado por una rejilla de dos capas de tiras de electrodos, una vertical y otra horizontal, separadas por un aislante y conectadas a un sofisticado circuito. El circuito se encarga de medir la capacidad mutua entre cada electrodo vertical y cada electrodo horizontal. Un dedo situado cerca de la intersección de dos electrodos modifica la capacidad mutua entre ellos al modificarse las propiedades dieléctricas de su entorno. El dedo tiene unas propiedades dieléctricas muy diferentes a las del aire.

La posición del dedo se calcula con precisión basándose en las variaciones de la capacidad mutua en varios puntos hasta determinar el centroide de la superficie de contacto. La resolución de este sistema es de hasta 1/40 mm. Además, se puede medir también la presión que se hace con el dedo. No se pueden usar lápices u otros materiales no conductores como punteros. Es muy resistente al entorno, soporta perfectamente polvo, humedad, electricidad estática, etcétera. Por otro lado, es ligero, fino y puede ser flexible o transparente.

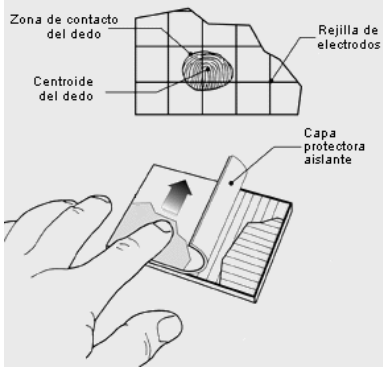


Figura 80

12-Sensor de huella digital

Es un dispositivo capaz de leer, almacenar e identificar las huellas dactilares. Su función básica es la de identificar personas para brindar accesos a lugares físicos o sistemas informáticos. Para identificar la persona primero debe asociarse a ésta un registro digital de la huella dactilar y almacenarla. Luego, para controlar el acceso se debe tomar nuevamente un registro digital y compararlo con los registros almacenados hasta darse con uno coincidente.

Existen una variada cantidad de tecnologías disponibles para realizar el registro digital de la huella dactilar:

a-Opticos reflexivos

Se basan en colocar el dedo sobre una superficie de cristal o un prisma que está iluminado por un diodo led. Cuando las crestas de las huellas del dedo tocan la superficie, la luz es absorbida, mientras que entre dichas crestas se

produce una reflexión total. La luz resultante y las zonas de oscuridad son registradas en un sensor de imagen como una técnica fotográfica digital.

b-Opticos transmisivos

La diferencia con el anterior radica en que el sensor ve una superficie más profunda y produce una imagen multiespectral. El uso de diferentes longitudes de onda para generar imágenes proporciona información de diferentes estructuras subcutáneas que permiten identificar de que el objeto en cuestión es un dedo genuino.

c-Capacitivos

El sensor es un circuito integrado cuyo objetivo es detectar las diferentes descargas electrostáticas entre las crestas y los valles de la huella dactilar. En este caso el dedo debe ejercer una pequeña presión sobre el circuito integrado para poder identificar las huellas.

d-Térmicos

En este caso se detecta el calor conducido por el dedo, el cual es mayor cuando hay una cresta que cuando hay un valle. Se ha desarrollado un componente de silicio con una matriz de píxeles cada uno de los cuales está cubierto con una capa de material piroeléctrico en el que un cambio de temperatura se traduce en un cambio en la distribución de carga de su superficie.

Capítulo VI
Monitores e impresoras

1-Monitores

El monitor de computadora es un visualizador que muestra al usuario los resultados del procesamiento de una computadora mediante una interfaz hombre-máquina basada en caracteres alfanuméricos e imágenes.

Aspectos singulares de los monitores en general

Píxel: unidad mínima representable en un monitor. Es un parámetro que mide la nitidez de la imagen, midiendo la distancia entre dos puntos; resulta fundamental a grandes resoluciones. Los tamaños de punto más pequeños producen imágenes más uniformes.

Área útil: el tamaño de la pantalla no coincide con el área real que se utiliza para representar los datos.

Ángulo de visión: es el máximo ángulo con el que puede verse el monitor sin que se degrade demasiado la imagen. Se mide en grados.

Luminancia: es la medida de luminosidad, medida en candela.

Tiempo de respuesta: también conocido como latencia. Es el tiempo que le cuesta a un píxel pasar de activo (blanco) a inactivo (negro) y después a activo de nuevo.

Contraste: es la proporción de brillo entre un píxel negro a un píxel blanco que el monitor es capaz de reproducir. Algo así como cuántos tonos de brillo tiene el monitor.

Coeficiente de contraste de imagen: se refiere a lo vivo que resultan los colores por la proporción de brillo empleada. A mayor coeficiente, mayor es la intensidad de los colores (30000:1 mostraría un colorido menos vivo que 50000:1).

Consumo: cantidad de energía consumida por el monitor, se mide en watts.

Hz o frecuencia de refresco vertical: es la velocidad en que se dibujan las líneas verticales de la imagen, y permite valorar la estabilidad y suavidad de transición de las imágenes.

Hz o frecuencia de refresco horizontal: similar al anterior, pero en sentido horizontal.

Tamaño de la pantalla: el tamaño de la pantalla es la distancia en diagonal de un vértice de la pantalla al opuesto (Ver Figura 81).

Figura 81

La resolución de pantalla: es el número de píxeles que pueden ser mostrados en una imagen. Se establece en ancho y alto.

Las resoluciones más usadas son:

SVGA	4:3	800	600
WSVGA	17:10	1024	600
XGA	4:3	1024	768
XGA+	4:3	1152	864
WXGA	16:9	1280	720
WXGA	5:3	1280	768
WXGA	16:10	1280	800
SXGA– (UVGA)	4:3	1280	960
SXGA	5:4	1280	1024
HD	16:9	1360	768
HD	16:9	1366	768
WXGA+	16:10	1440	900
HD+	16:9	1600	900
UXGA	4:3	1600	1200
WSXGA+	16:10	1680	1050
FHD	16:9	1920	1080
WUXGA	16:10	1920	1200

WQHD	16:9	2560	1440
QFHD	16:9	3840	2160

En SMART TV se pueden observar las siguientes resoluciones:

Nombre de resolución	Pixeles horizontales x Verticales	Otros nombres	Dispositivos
8K	7,680 x 4,320	ninguno	Televisores conceptuales
"Cinema" 4K	4,096 x [sin especificar]	4K	Proyectores
UHD	3,840 x 2,160	4K, Ultra HD, Definición Ultra Alta	Televisores
2K	2,048 x [sin especificar]	ninguno	Proyectores
WUXGA	1,920 x 1,200	Widescreen Ultra Extended Graphics Array	Monitores, proyectores
1080p	1,920 x 1,080	Full HD, FHD, HD, Alta Definición, 2K	Televisores y monitores
720p	1,280 x 720	HD, Alta Definición	Televisores

1.1-Monitor de LCD

Los LCD (del inglés, *Liquid Cristal Display* o monitor de cristal líquido), se fundamentan en el aprovechamiento de determinadas características de los cristales líquidos.

En la materia en estado sólido, las moléculas tienden a mantener su orientación y posición siempre de la misma forma. El estado líquido se caracteriza porque las moléculas cambian su orientación. El cristal líquido es una sustancia que tiene un estado raro, en el cual las moléculas mantienen su orientación, pero pueden moverse a otras posiciones.

Los LCD se basan en tres principios:

- La luz se puede polarizar.
- Las moléculas de los cristales líquidos pueden cambiar de orientación por inducción eléctrica para, de esta manera, permitir el paso de la luz o no.

- Hay sustancias transparentes que pueden conducir electricidad, como el óxido de estaño e indio (ITO).

Cristales polarizados

Los cristales sólidos pueden polarizarse, lo que les confieren una serie de propiedades de aplicación tecnológica de uso en los paneles LCD.

La luz, como onda electromagnética, está compuesta de un campo eléctrico y un campo magnético que oscilan transversalmente a la dirección de propagación de la misma. En un rayo de luz natural, las diferentes ondas que lo componen (al oscilar a diferentes longitudes y fases) provocan que no haya una dirección determinada del campo eléctrico resultante. Se dice en este caso que la luz no está polarizada.

Pues bien, los cristales polarizadores «filtran» la luz de manera que solo dejan pasar aquellas ondas que se correspondan con ángulo específico de vibración. La luz que pasa al otro lado es luz polarizada y solo contendrá ondas de un determinado ángulo.

En la figura 82, se muestran dos cristales polarizados: en el primer ejemplo, los dos cristales tienen su polarización orientada en el mismo sentido; en el segundo ejemplo, los dos cristales tienen su polarización orientada en sentidos contrarios. Por lo tanto, el primer ejemplo permite que la luz traspase los dos cristales, mientras que en el segundo ejemplo la luz no puede pasar el segundo cristal.

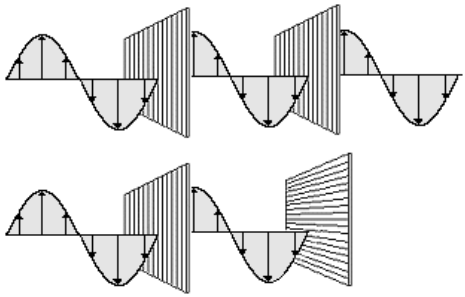


Figura 82

Movimiento molecular

El cristal líquido tiene unas moléculas alargadas que pueden reorientarse con la incidencia de corriente eléctrica y de esta forma reorientar la onda luz.

Funcionamiento de los LCD activos con retroiluminación

Este es el tipo de LCD más utilizado actualmente y el que requiere de mayor explicación. Los LCD pasivos y los reflectivos son más simples y se fundamentan en el mismo principio de manejo del cristal líquido.

La formación de cada píxel de un LCD requiere de:

1. Una fuente de luz homogénea en su parte posterior (hoy se está utilizando la tecnología de led para esta tarea).
2. Dos vidrios polarizados colocados como en el ejemplo anterior uno a 90 grados del otro (esto es a modo de explicación, ya que desde luego que no es un cristal para cada píxel, sino que se utiliza el mismo cristal para todos los pixeles del monitor).
3. Entre ellos se colocan varias capas de cristal líquido.
4. Entre las capas de cristal líquido y los vidrios polarizados, se colocan cristales con los electrodos transparentes soldados sobre el cristal (una capa por delante de los cristales líquidos y otra por detrás).
5. Si los electrodos no emiten corriente, la orientación de las moléculas del cristal líquido no alteran el ángulo de dirección de la luz; por lo tanto, la luz trasera no podrá pasar el segundo vidrio polarizado.
6. Si los electrodos inducen corriente sobre el cristal líquido, hacen que las moléculas del cristal se reorienten de tal forma que estas moléculas, a su vez, giren el ángulo de la luz 90 grados para que pueda superar el segundo cristal polarizado.

Este tipo de LCD depende de unos transistores especiales llamados en inglés *thin film transistors* (TFT). Son pequeños transistores y condensadores que trabajan como un switch y se sueldan en forma de matriz en un sustrato de vidrio. La estructura interna de un LCD de matriz activa se muestra en la figura 83.

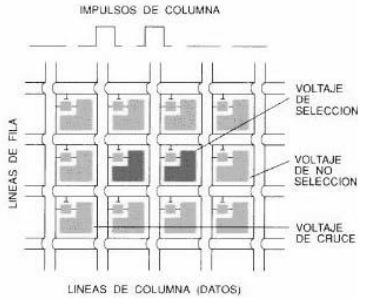


Figura 83

Por selección de fila y columna, se activan las compuertas de los transistores para energizar el cristal líquido. Los transistores se colocan en una esquina de cada píxel, por lo que al ser poco trasparentes, oscurecen un poco el monitor.

Aplicación de color

Por delante del segundo cristal polarizado, se coloca un cristal de color (rojo, azul o verde). Cada píxel del monitor estará formado por uno de los tres colores. Por lo tanto, para formar un único píxel (de imagen), se requiere de tres píxeles. Como los tres están muy juntos, el ojo humano lo ve como uno solo (Ver Figura 84).

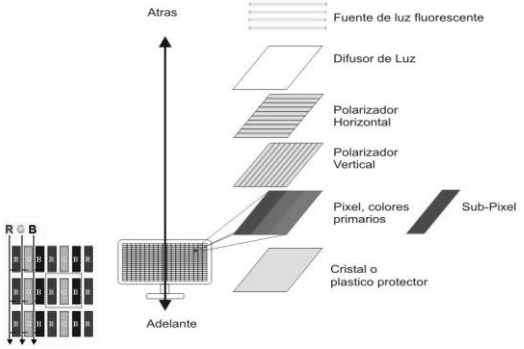


Figura 84

Entre los polarizados de la imagen anterior, se ubican los cristales con los electrodos y las capas con el cristal líquido que reorientan la luz, como se puede ver en la figura 85.

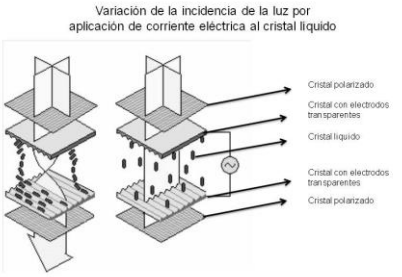


Figura 85

1.2-Monitor de plasma

El funcionamiento de un monitor de plasma (PDP: *Plasma Display Panel*) es, en cierta forma, bastante similar al LCD. Consta de dos capas de hilos conductores transparentes con filas perpendiculares entre ellas. Estas se hallan pegadas a sendas caras de unos sustratos de vidrio que encierran un gas (normalmente neón). Al producirse suficiente corriente en las intersecciones, el gas pasa a estado de plasma, emitiendo luz.

1.3-Pantallas táctiles

Es una pantalla que mediante un toque directo sobre su superficie permite la entrada de datos y órdenes al dispositivo. Este contacto también se puede realizar por medio de un lápiz óptico u otras herramientas similares.

Hay dos tipos de pantallas táctiles:

Resistivas (Ver Figura 86): son más baratas y no les afectan el polvo ni el agua, y además pueden ser usadas con un puntero (preferentemente) o con el dedo. Sin embargo, pierden hasta un 25% del brillo.

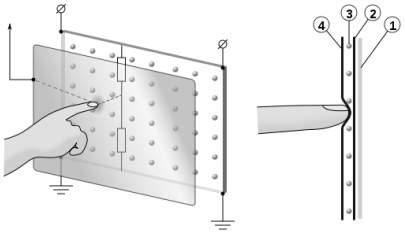


Figura 86

Están integradas por dos capas (4 y 2) con un espacio de separación entre ellas (3). La capa 1 es un cristal que separa la pantalla táctil del monitor (LCD o plasma). Cuando se presiona la capa superior (4), se calcula la tención en un eje de coordenadas; y cuando la presión llega a la segunda (2) capa, se calcula la tención sobre el otro eje de coordenadas. Cada capa conductora está tratada con un material conductor resistivo transparente, normalmente óxido de indio y estaño (In2O3) (SnO2).

Capacitivas (Ver Figura 87): la calidad de imagen es mejor, son mucho más precisas y permiten el uso de varios dedos a la vez (*multitouch*). Sin embargo, son más caras y no se pueden usar con puntero normal, sino con uno especial para las pantallas capacitivas.

En estas pantallas, se añade una capa conductora (normalmente del mismo material que la que utilizan las resistivas) al cristal del propio monitor. Esta capa almacena cargas que se sitúan sobre el cristal del monitor. Cuando un usuario toca el monitor, algunas cargas se transfieren al usuario, de tal forma que la carga en la capa capacitiva se reduce. Este decrecimiento se mide en los circuitos situados en cada esquina del monitor. El ordenador calcula, por la diferencia de carga entre cada esquina, el sitio concreto donde se tocó y envía la información al software de control de la pantalla táctil.

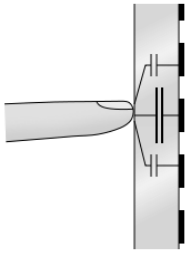


Figura 87

2-Impresoras

Una impresora es un periférico de salida, que permite producir una copia de un texto o gráficos, almacenado en formato electrónico, sobre papel o transparencia.

2.1-Impresoras de impacto

Las impresoras de impacto se basan en la fuerza de impacto para transferir tinta al medio, de manera similar a las máquinas de escribir. Fueron las primeras que surgieron en el mercado, y aunque han perdido protagonismo frente a la impresora de inyección o la impresora láser, siguen siendo muy útiles para la impresión de formularios continuos. Las impresoras de impacto están limitadas a reproducir texto en blanco y negro.

Los tipos aún utilizados de impresoras de impacto son:

- Impresora de margarita (Ver Figura 88): contiene todos los caracteres distribuidos radialmente en una rueda. La rueda gira posicionando el caracter que se desea imprimir frente a la cinta. Un martillo golpea el carácter contra la cinta, transfiriéndose así la tinta al papel.

Figura 88

- Impresora de bola (Ver Figura 89): contiene todos los caracteres en una esfera que gira sobre un soporte móvil hasta colocar el caracter deseado frente a la cinta y golpearla para imprimir el carácter en el papel.

Figura 89

- Impresora de línea (Ver Figura 90): tiene una cadena (con todos los caracteres necesarios y repetidos) que gira a gran velocidad. Los caracteres son impactados por los múltiples martillos cuando pasan por el lugar correcto. De esta forma, es capaz de realizar las impresiones por líneas y no por caracteres, lo que implica un gran aumento en la velocidad de impresión.

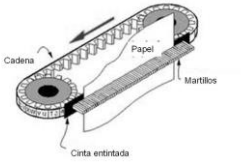


Figura 90

- Impresora matricial (Ver Figura 91): la impresión se realiza mediante unos cabezales que contienen una matriz de píxeles o de puntos. El cabezal contiene unas agujas (entre 7 y 24) que son impulsadas contra la tinta, formándose los caracteres mediante pequeños puntos. Esto permite imprimir diferentes fuentes e incluso imágenes

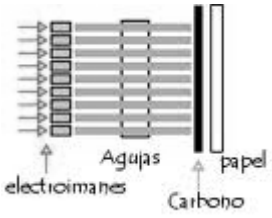


Figura 91

2.2-Impresoras láser (Ver Figura 92):

Las impresoras de láser trabajan utilizando el principio de la xerografía, que es la utilizada en la mayoría de las fotocopiadoras. Funcionan adhiriendo tóner a un tambor de impresión sensible a la luz y utilizando la electricidad estática para transferir el tóner al medio de impresión (papel o transparencia) al cual se une gracias al calor y la presión.

Las impresoras láser son conocidas por su impresión de alta calidad, buena velocidad de impresión y su bajo costo por copia. Son las impresoras más comunes para muchas de las aplicaciones de oficina de propósito general. Son menos utilizadas por el consumidor debido a su alto coste inicial. Las impresoras láser están disponibles tanto en color como en monocromo.

Las etapas de la impresión láser son las siguientes:

- La impresora recibe la orden desde la computadora de lo que va a imprimir.
- La impresora almacena los datos recibidos en la memoria RAM (de la impresora) también llamada Buffer.
- Un mecanismo electromecánico acomoda la hoja acorde a las especificaciones que envía la computadora.
- Un mecanismo llamado escáner emite un haz de luz láser que se refleja con un espejo sobre el cilindro fotorreceptor. Este haz de luz lleva cargas electroestáticas, las cuales atraen el polvo de tinta para forman el caracter o figura.
- El fotorreceptor gira y se carga de tóner para luego pasar sobre la hoja. La tinta en polvo se sobrepone en la hoja.
- Luego la hoja pasa sobre un mecanismo llamado fusor, el cual gira y, además, está caliente, por lo que se derrite la tinta en polvo, y una vez que se enfría, la tinta está pegada en la hoja.

- La hoja va avanzando por medio de un rodillo movido por un motor, conforme se termina de imprimir cada renglón, se mueve para empezar el siguiente.
- Lo anterior se repite hasta terminar los datos almacenados.

Normalmente, la resolución en las impresoras a láser es medida en términos de dpi (puntos por pulgada). Cuanto mayor que sea el número de dpi, más nítida será la imagen.

Para imprimir textos, es suficiente una resolución de 300 dpi. Para imágenes es necesario al menos 600 dpi de resolución.

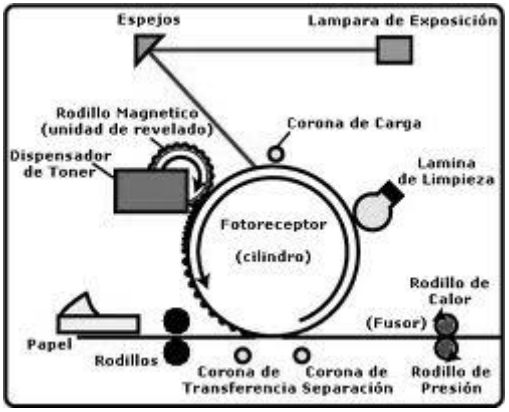


Figura 92

2.3-Impresora de chorro de tinta

Las impresoras de inyección de tinta (*Ink Jet*) rocían hacia el medio cantidades muy pequeñas de tinta, usualmente unos picolitros. Para aplicaciones de color, incluyendo impresión de fotos, los métodos de chorro de tinta son los dominantes, ya que las impresoras de alta calidad son poco costosas de producir y son, adicionalmente, capaces de imprimir en papel fotográfico.

El método de estas impresoras consiste en inyectores que producen burbujas muy pequeñas de tinta que se convierten en pequeñísimas gotitas de tinta. Los puntos formados son los pequeños pixeles.

Existen dos métodos para inyectar la tinta:

- Método térmico: un impulso eléctrico generado por un semiconductor produce un aumento de temperatura (aproximadamente 480 °C durante microsegundos) que hace hervir una pequeña cantidad de tinta dentro de una diminuta cámara (parecida a una pequeña aguja de una jeringa) y forma una burbuja de vapor que al expandirse empuja una minúscula gota de tinta sobre el papel (Ver Figura 93). Este método tiene el inconveniente de limitar en gran medida la vida de los inyectores, es por eso que estos inyectores se encuentran en los cartuchos de tinta.

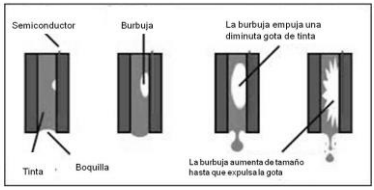


Figura 93

- Método piezoeléctrico: cada inyector está formado por un elemento piezoeléctrico (cristal) que al recibir un impulso eléctrico vibra generando un brusco cambio de presión en el interior del cabezal provocando la inyección de una partícula de tinta. Su ciclo de inyección es más rápido que el térmico. Las cabezas de impresión piezoeléctricas pueden utilizar tinta que se seca con mayor rapidez y pigmentos que podrían dañarse con las temperaturas en una cabeza térmica. Asimismo, como una cabeza piezoeléctrica está integrada a la impresora, solo se necesita reemplazar el cartucho de tinta (Ver Figura 94).

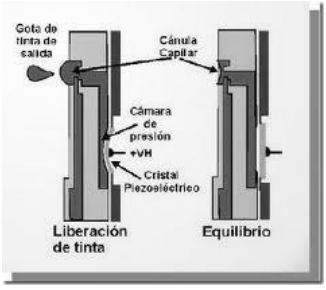


Figura 94

Capítulo VII
Tipos de computadoras

1-Estaciones de trabajo/computadoras personales

1.1-Desktop

Desktop significa literalmente “parte superior de un escritorio”, pero para aclarar las cosas, normalmente se le dice computadora de escritorio o computadora de escritorio. Fueron los primeros tipos de computadoras personales o PC y claro que no están diseñados para la movilidad. Están diseñadas para ser utilizadas en un lugar fijo, generalmente en el escritorio y por eso su nombre.

Hoy están ofreciendo las siguientes capacidades:

> Velocidad del procesador: 3,2 GHz. Cache: 12 MB. Núcleos: 6 de 2 hilos por núcleos. Capacidad de memoria RAM: 128 GB. Capacidad de almacenamiento secundario: 2 TB de 7200 RPM. Unidad Óptica: DVD. Puertos USB: 7. Teclado. Mouse. Parlantes. Tarjeta de red.

1.2-Notebook

Su significado literal es por su semejanza con ellos. En general se les dice portátiles y son computadoras móviles que integran una pantalla, un teclado, un touchpad, el procesador, la memoria y el disco duro todo en un paquete con una batería.

Hoy están ofreciendo las siguientes capacidades:

> Velocidad del procesador: 3,2 GHz. Cache: 12 MB. Núcleos: 8 de 2 hilos por núcleos. Capacidad de memoria RAM: 32 GB. Capacidad de almacenamiento secundario: 1 TB de 7200 RPM. Unidad Óptica: DVD. Puertos USB: 4. Tarjeta de red.

1.3-Tablet

La Tablet es una computadora portátil de mayor tamaño que los teléfonos inteligentes. Tiene una pantalla táctil mediante la cual se interactúa utilizando básicamente los dedos, por lo que no hay necesidad de tener un teclado físico ni tampoco un ratón o mouse. Habitualmente sus pantallas son de 7 a 12 pulgadas y son muy livianas. Todas esas características las hacen muy fáciles de transportar.

Hoy están ofreciendo las siguientes capacidades:

> Velocidad del procesador: 2 GHz. Núcleos: 8. Capacidad de memoria RAM: 3 GB. Capacidad de almacenamiento secundario: MicroSD 256 GB. Puertos USB: 2. Wi-Fi. Bluetooth. Resolución de pantalla de 1536 X 2048 Pixeles. Peso: 400 g.

1.4-Smartphone

El teléfono inteligente (smartphone en inglés) es un tipo de ordenador de bolsillo que combina los elementos de una tablet, un teléfono móvil, una cámara de fotos y una cámara de video.

Hoy están ofreciendo las siguientes capacidades:

> Velocidad del procesador: 2,8 GHz. Núcleos: 8. Capacidad de memoria RAM: 4 GB. Capacidad de almacenamiento secundario: 256 GB. Puertos USB: 1. Wi-Fi, GPS, Bluetooth, Pantalla de 8 pulgadas de 2960 x 1.440 pixeles. Cámara de 12 MP.

2-Servidores

Un servidor es una computadora que ha sido optimizada para proporcionar servicios a otras computadoras a través de una red. Normalmente los servidores tienen procesadores potentes, gran cantidad de memoria y mucha capacidad de almacenamiento. Pueden tener diversos usos como servidores web, servidores de correo electrónico, servidores de aplicaciones para empresas, etc.

> Actualmente Google esta utilizando servidores de 2 procesadores x86 que pueden llegar hasta velocidades de 4 GHz, de 16 núcleo. Con 8 slots para memoria RAM de hasta 192 GB. Capacidad de almacenamiento de 2 discos duros de 30 TB cada uno. Cada uno de estos servidores están instalados en contenedores de 1.160 servidores. No se conoce que cantidad de estos contenedores dispone en sus diversas instalaciones.

Los servidores que se comercializan para empresas tienen las siguientes características:

> Velocidad del procesador: 3,5 GHz. Núcleos: 14. Capacidad de memoria RAM: 256 GB. Capacidad de almacenamiento secundario: 4 TB. Controladores RAID. Puertos USB: 8. Conexiones de red: 4.

3-Supercomputadoras

Se llaman asía a las computadoras con más capacidad de cálculo existentes. Este tipo de computadoras están destinadas, principalmente, a actividades científicas.

El ejemplo de este tipo es la supercomputadora es la china Sunway TaihuLight. Tiene un rendimiento de 125.436 TFlops/s (que es como se miden este tipo de computadoras). Esta medida está relacionada con la cantidad de operaciones matemáticas de punto flotante que pueden hacer por segundo. Tiene una memoria RAM de 1.310.720 GB y su procesador esta formado por 10.649.600 núcleos.

FERNANDO J. MARTINI

TECNOLOGIA DE LA INFORMACION

VOLUMEN 2

EL SOFTWARE Y EL DBMS

TECNOLOGÍA DE LA INFORMACIÓN
VOLUMEN 2

EL SOFTWARE Y EL DBMS

Versión: 1.02

Martini, Fernando J.
Tecnologia de la Información : el software y el DBMS . - 1a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2014.
v.2, CD-ROM.

ISBN 978-987-33-4693-4

1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

Martini, Fernando J.
Tecnología de la Información. - 1a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2014.
v.3, CD-ROM.
ISBN 978-987-33-4691-0
1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

ÍNDICE

Capítulo I
El software del sistema

1-El software

A pesar de los numerosos esfuerzos realizados hasta ahora en el ámbito de la computación, los científicos y técnicos no han logrado aún encontrar la forma de crear máquinas capaces de aprender a resolver problemas por ellas mismas. Esto implica que para que las computadoras presten un buen servicio a las tareas de los hombres, es necesario brindarles las instrucciones detalladas para su operación, o sea, programarlas. A los programas de computadora, técnicamente se los llama software.

El software se clasifica en dos grandes grupos:

- De sistemas: es el conjunto de programas que permiten utilizar los recursos físicos de la computadora (hardware). Este, a su vez, puede subdividirse de la siguiente forma:

    - El sistema operativo (de ahora en más SO), que permite simplificar a las aplicaciones el uso del hardware, temporizar sucesos, asignar recursos y monitorear eventos. También se lo denomina Kernel o modo supervisor. Este conjunto de programas está protegido por el hardware para que el usuario no pueda acceder a modificarlo. Los recursos de las computadoras se comparten entre distintos usuarios o tareas, por lo tanto, el SO debe hacer de mediador o arbitraje en la asignación de recursos y en la asignación de tiempo para su utilización.

      El monitoreo de eventos consiste en poder controlar el funcionamiento del uso de memoria, la cantidad de accesos a los discos, los usuarios en actividad, la cantidad de programas en ejecución, la identificación de cada programa, las velocidades de acceso a memoria y a disco, etcétera.

Por encima de este, se encuentran:

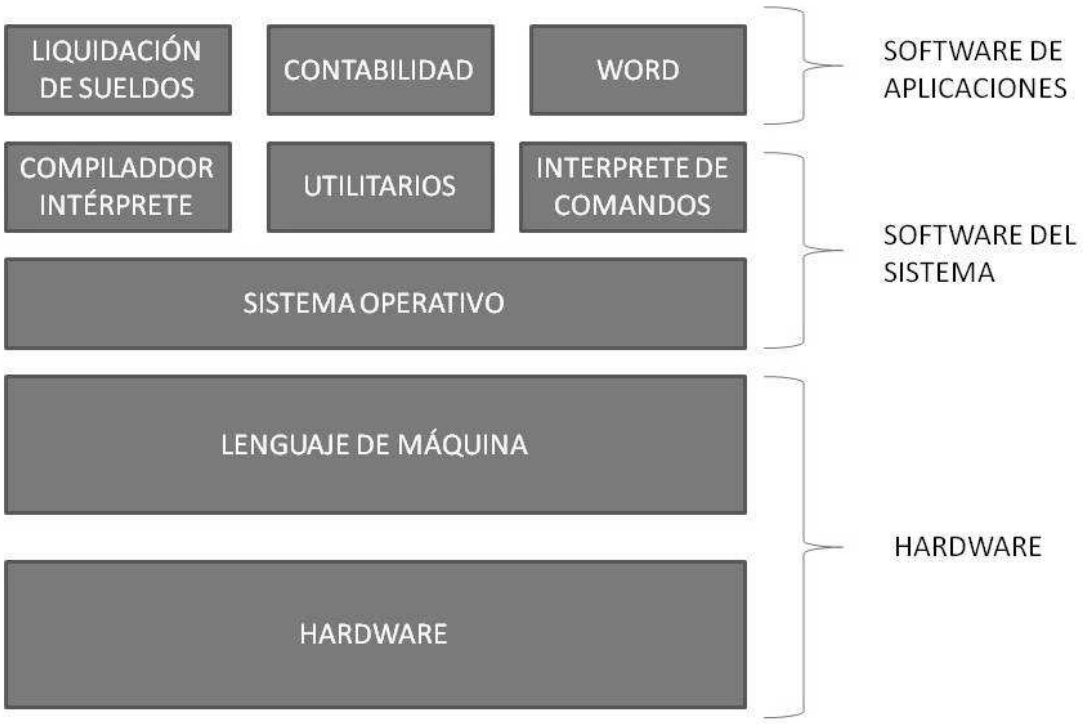
- **Programas utilitarios** que se utilizan para realizar tareas rutinarias, como copiar, crear, desfragmentar archivos, realizar backup, editar, etcétera.

- **Intérprete de comandos** o también llamado *Shell*.

- **Compiladores o traductores** de lenguajes, cuya finalidad es convertir las instrucciones escritas en lenguaje de alto nivel (COBOL, BASIC, Pascal, etcétera.) a lenguaje de máquina.

- De aplicaciones: se refiere a los programas que han sido escritos con el fin de utilizar la computadora para una función específica y destinada al usuario final. Por ejemplo: Liquidación de sueldos, contabilidad, word, etcétera.

Se puede graficar de la siguiente forma:



Esta gráfica contiene aspectos aún no mencionados que se verán a continuación.

2-El lenguaje de máquina y el ensamblador

Para que la computadora entienda nuestras instrucciones, se debe usar un lenguaje específico, conocido como código de máquina, que la máquina comprende fácilmente, pero que lo hace excesivamente complicado para las personas. De hecho, solo consiste en cadenas extensas de números 0 y 1. El lenguaje de máquina fue el primero que se empleó para la programación de las primeras computadoras. Una instrucción en lenguaje de máquina es algo parecido a lo siguiente: 011011001010010011110110 (esta es una instrucción para trasladar el contenido de la posición de memoria X a la posición de memoria Y).

Por ejemplo, una instrucción en lenguaje de alto nivel java, como el siguiente: contador = contador + 1; se representaría en lenguaje de máquina como lo siguiente: 00000111010100011111000010001000010101010. Este conjunto de unos y ceros no solamente hacen dificultosa la interpretación de una sola instrucción, sino que la comprensión de un programa sería prácticamente imposible.

El lenguaje de máquina está directamente relacionado al microprocesador que se quiera programar. Un lenguaje de máquina preparado para un tipo de procesador no podrá ser utilizado en otro procesador, salvo que sea un procesador de la misma familia o línea. Este conjunto de instrucciones de máquina son las únicas que reconoce el microprocesador y, por así decirlo, están grabadas o «alambradas» en el hardware y no pueden ser cambiadas. El fabricante que diseñó el micro ya preparó ese conjunto de instrucciones dentro del hardware (mediante condensadores, transistores, diodos, etcétera) y estableció un código binario para su utilización.

Con el fin de simplificar las dificultades que el código de máquina genera para poder programar las computadoras, se pensó en utilizar las bondades de la misma computadora para ayudar en esta tarea. Por lo tanto, se pensó en un conjunto de instrucciones alfabéticas que correspondan unívocamente con cada instrucción binaria. A estas instrucciones alfabéticas, se las denominaron mnemotécnicos.

Por ejemplo: la instrucción binaria expuesta anteriormente mediante, la cual se puede trasladar el contenido de la posición de memoria **x** a la posición de memoria **y**, escrita con instrucciones en mnemotécnicos, sería igual a «TRASLADAR 11010110 00011101». A partir de estos mnemotécnicos, se comenzaron a construir programas

ensambladores, cuya finalidad primordial era traducir el código alfabético al código binario. En el ejemplo, TRASLADAR = 00000111 (el primer byte de la instrucción).

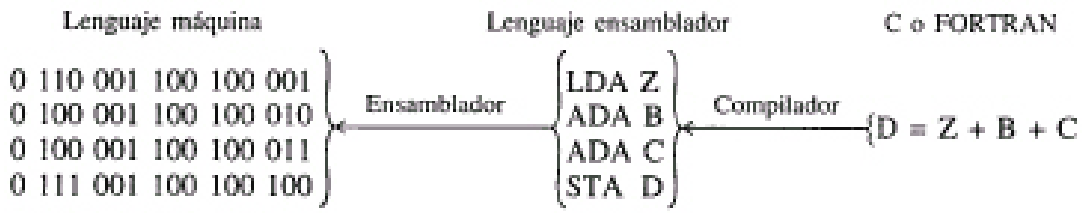
Con el avance de los programas ensambladores, se comenzaron a utilizar otras ventajas como, por ejemplo, asignar un nombre a la dirección de memoria en lugar de tener que asignarla con un valor binario. La instrucción anterior quedaría, por ejemplo, de la siguiente forma: TRASLADAR POS-A POS-B. Esto dio lugar a la aparición de lenguajes ensambladores que son traducidos por el programa ensamblador.

El lenguaje ensamblador siempre tiene una correspondencia unívoca con el lenguaje de máquina, es decir que cada instrucción binaria tiene su correspondiente instrucción en lenguaje ensamblador.

3-El compiladores

A pesar de la construcción del lenguaje ensamblador y de los programas ensambladores, la programación de un microprocesador continuaba siendo muy compleja. Por tal motivo, se pensó en desarrollar un lenguaje aún más comprensible y simple para los humanos. A estos lenguajes, se los llamaron lenguajes de alto nivel. Entre los lenguajes de alto nivel más conocidos, se encuentran el Pascal, Fortran, COBOL, C, C++, JAVA, etcétera.

Para traducir los programas de alto nivel a uno de bajo nivel, se requiere de un programa denominado compilador. A los programas de alto nivel, se los suele llamar programas fuentes; y a los de bajo nivel, programas objeto. El objetivo del lenguaje de alto nivel es que los programadores de aplicaciones puedan programar con un conjunto reducido de instrucciones. Normalmente, a una instrucción de alto nivel le corresponden varias de bajo nivel. En el siguiente ejemplo, se puede ver cómo unas instrucciones del lenguaje de alto nivel C o FORTRAN se convierten en lenguaje de bajo nivel o ensamblador y luego a lenguaje de máquina:



4-El enlazador también llamado *linker*

Para que un programa objeto quede completo, se requiere que sea pasado por un programa llamado enlazador (ligado o *linker*). Este proceso tiene tres funciones muy importantes:

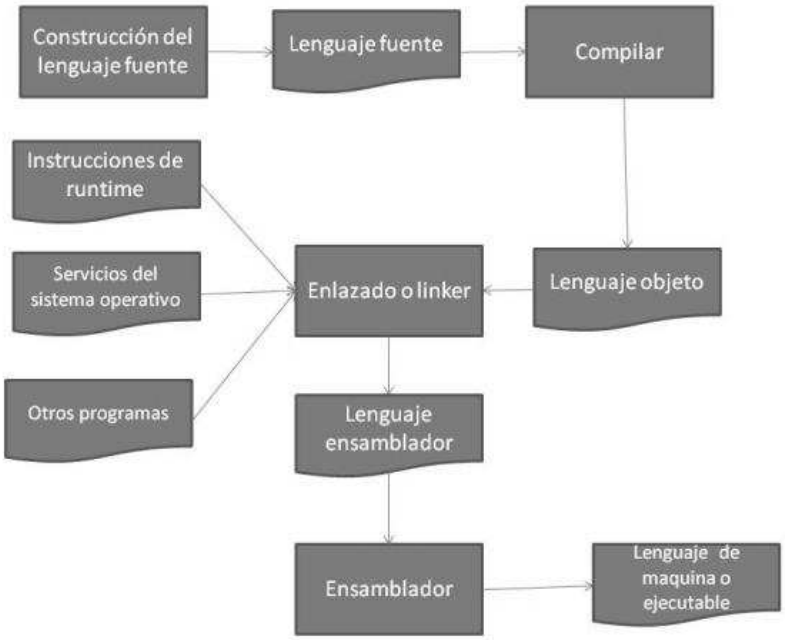
a) Agregar al programa objeto las instrucciones denominadas de *run-time*. Estas son un conjunto de instrucciones muy técnicas (manejo de memoria, control de errores, etcétera) para que el programa pueda ser ejecutado por el procesador.

b) Agregar las instrucciones del servicio del sistema operativo, no hay que olvidar que toda aplicación debe estar permanentemente controlada por el sistema operativo.

c) Enlazar el programa compilado con otros programas objetos construidos con anterioridad. Se construyen muchos programas para ser reutilizados en otros con el objetivo de no tener que volver a codificar nuevamente lo mismo. Por ejemplo: las instrucciones para calcular una fecha a partir de otra fecha, a la cual hay que sumarle una cantidad de horas, es un conjunto de instrucciones muy utilizadas en los programas. Por lo tanto, este programa se construye de tal forma que pueda ser invocado por otros que lo requieran. El enlazador es el programa que une estos programas reutilizables.

5-El proceso completo de generar una aplicación compilable

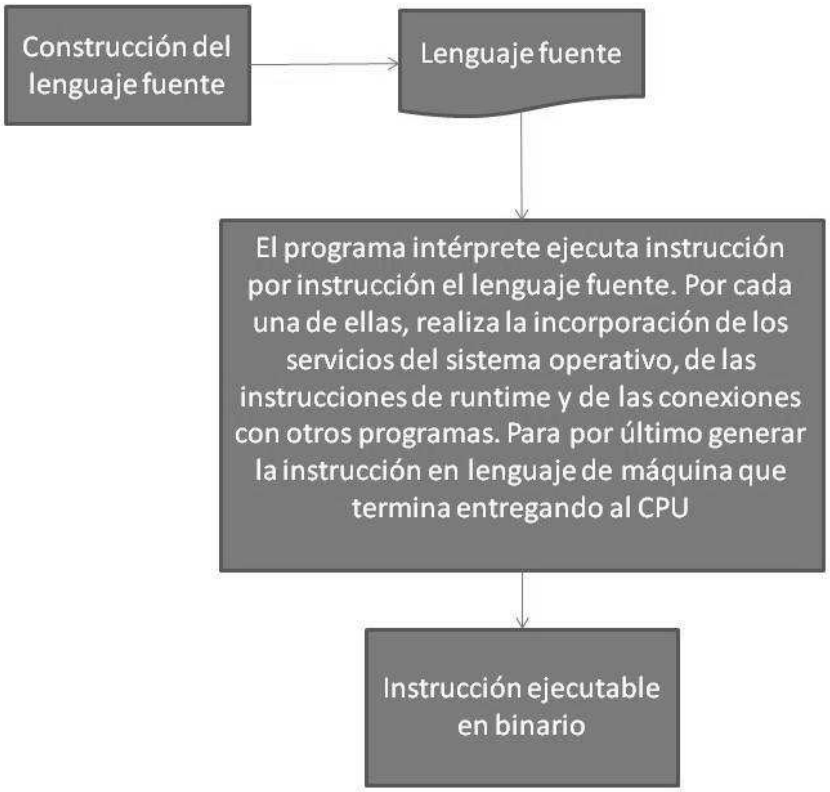
Hoy en día, el proceso de generar un programa aplicativo requiere que se efectúe el siguiente procedimiento:



a) Se construye el programa fuente en algún lenguaje de alto nivel.
b) Se compila y se obtiene el programa objeto en lenguaje ensamblador.
c) El programa objeto se enlaza con la librería de run-time, los servicios del sistema operativo y otros programas requeridos por la aplicación. De esta forma, se obtiene el programa objeto enlazado.
d) El programa objeto enlazado se traduce con el ensamblador a lenguaje de máquina generando el programa ejecutable.

6-El intérprete

El intérprete, a diferencia del compilador, es un programa que permite ejecutar los programas desde su instrucción de alto nivel, como DBase, BASIC, APL, Lisp, etcétera. Para que se pueda ejecutar el programa de alto nivel, es necesario que el intérprete tome cada instrucción de alto nivel e inmediatamente la convierta en una instrucción de máquina. A continuación, se mostrará una gráfica que explica el funcionamiento del programa intérprete:



En general, la principal desventaja de los intérpretes es que, cuando un programa es interpretado, suele ejecutarse más lento que si el mismo programa estuviese compilado. Esto se debe a que el intérprete debe analizar cada sentencia del programa en cada ejecución (un análisis en tiempo real). También, el acceso a variables es más lento en un intérprete, porque mapear los identificadores para almacenar las localizaciones se debe hacer repetidas veces en tiempo real.

La ventaja del trabajar con programas interpretados es que no se requiere guardar un doble juego de programas: por un lado, el programa fuente; y por otro, el programa ejecutable.

En general, todos los lenguajes pueden trabajar en forma compilada o interpretada si se lo quisiera. Para algunos lenguajes de alto nivel, no se conocen intérpretes; y para algunos lenguajes, no se conocen compiladores y solo se ejecutan en forma interpretada. Esto es una decisión comercial más que una imposibilidad técnica. La decisión comercial se puede deber a que en una forma u otra el programa trabaja en muy baja performance.

7-Intérprete de comandos (*Shell*)

El intérprete de comandos, también denominado *shell*, se emplea para referirse a un tipo especial de programa que provee una interfaz de usuario para acceder a los servicios del sistema operativo. Estos pueden ser gráficos o de texto simple, dependiendo del tipo de interfaz que empleen. Por ejemplo: el *shell* es el programa desde donde pueden ejecutarse copias de archivos, crear archivos o directorios, borrar archivos o directorios, dar formato o reorganizar un disco, crear permisos de accesos a los archivos y directorios, etcétera.

Capítulo II
El sistema operativo

1-Generalidades

Un SO es un conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, y que se ejecuta en modo privilegiado respecto de los restantes.

Una de las principales funciones de un SO es la de simplificar el uso del hardware, esta funcionalidad es la que le permite al programador de aplicaciones abstraerse de las complicaciones del hardware y utilizar una serie de llamadas al SO para que este sea el encargado resolver los aspectos físicos. Si no existiese esta funcionalidad, todo programador, por ejemplo, debería preocuparse por cosas como lo siguiente:

> Una lectura de un dato en un disco requiere de aproximadamente 13 parámetros de 9 bytes. Estos parámetros especifican cosas como el bloque del disco que se leerá, el número de pista o pistas, el modo de grabación, etcétera La lectura devuelve 23 campos de estado y errores. El programador debería hasta lidiar con cosas tan físicas como, en el caso de leer una unidad de floppy disk, verificar si el motor de la unidad está funcionando, porque de lo contrario deberá darle arranque y controlar que el motor no esté demasiado tiempo encendido, de lo contrario podrá dañar el floppy disk.

El SO suele tener entre 100 y 150 instrucciones o llamadas que permiten al programador de aplicaciones simplificar mucho su trabajo. Por ejemplo, y para seguir con el caso anterior, cuando la aplicación quiere leer un dato de un archivo, solo se deberá informar al SO algunas pocas cosas, por ejemplo: el nombre del archivo, su ubicación, el o los registros que se desean leer y muy pocas cosas más. La devolución del sistema operativo es un código que muestra si la lectura fue exitosa o no y por qué no. No tiene que hacer referencia a cosas como pistas, sectores, arranques de motores, etcétera.

2-BIOS

El BIOS (sigla en inglés de *basic input/output system*; en español «sistema básico de entrada y salida») es un tipo de firmware (programa grabado en una memoria tipo

ROM) que localiza y prepara los componentes electrónicos o periféricos de una máquina, para comunicarlos con algún sistema operativo que la gobernará. Para ello la máquina cargará ese sencillo programa en la memoria RAM central del aparato. El programa realizará el chequeo de la memoria principal y secundaria, la comunicación con el usuario (vía monitor o teclado) y el enlace con el núcleo del sistema operativo que gobernará el sistema.

Concretamente, las funciones del BIOS son:

a) Inventario y comprobación del hardware.
b) Inicialización de los dispositivos hardware que lo requieren (carga de cierto software básico).
c) Inicio del SO.

3-Procesos

Entender el concepto de proceso es importante en cualquier SO. Por lo tanto, se definirá proceso a un programa en estado de ejecución. El SO asigna un espacio de memoria a cada proceso para que se almacenen las instrucciones del programa en lenguaje de máquina, los datos que incorpora y los que genera. Es decir que el proceso es el programa más toda la información asociada respecto de su tarea específica.

3.1-El procesamiento en seudoparalelo - Multiprogramación

Para que se entienda bien el seudoparalelismo, es fundamental comprender el concepto de multiprogramación. La multiprogramación permite administrar varios procesos en una sola CPU, alternando la ejecución de estos a través de controles de tiempo, asignando límites temporales de ejecución a cada proceso y alternándolos a medida que este límite es alcanzado. La gran mayoría de las computadoras pueden realizar varias tareas al mismo tiempo. Mientras se ejecuta un programa, una computadora, además, puede estar leyendo de un disco y enviando texto a una pantalla o impresora. El concepto de multiprogramación incorpora la capacidad del SO para que la CPU conmute de un programa a otro, ejecutando cada uno durante decenas o centenas de milisegundos. Esta capacidad del SO y del hardware de hacer que se pueda conmutar rápidamente de un proceso a otro da al usuario la sensación de que la computadora está atendiendo muchos procesos en paralelo.

Por este motivo, un programa puede estar en alguna de las siguientes tres situaciones:

1. En ejecución (cuando la CPU le asigna tiempo para trabajar sobre ella).

2. Listo para ejecutar (en este estado, se mantiene hasta que la CPU le asigne su tiempo).

3. Bloqueado (esto ocurre porque el programa está esperando que ocurra algún evento externo, cuando este evento ocurra, pasará a estado de listo).

3.2-Jerarquía de procesos

Cuando un SO arranca, este crea algunos procesos principales que serán los encargados de crear todos los demás procesos. Como, a su vez, cualquier proceso puede crear otros procesos (denominados «hijos»), se obtiene una estructura jerárquica en forma de árbol donde todo proceso (excepto los iniciales del SO) tiene un padre (o antecesor) y, a su vez, tiene la capacidad de tener un hijo o más. A este conjunto conformado por un padre y todos sus hijos (y recursivamente los hijos de estos), se lo denomina familia de procesos (Ver Figura 1).

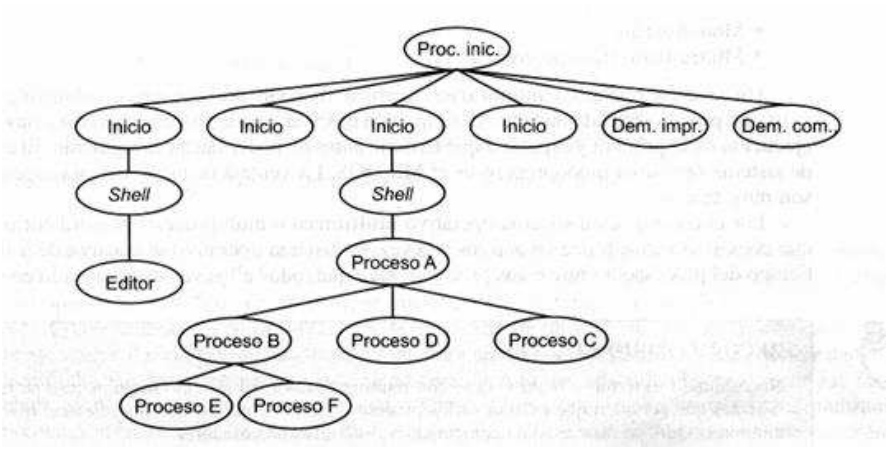


Figura 1

Para poder suspender y arrancar procesos en ejecución, el SO utiliza una tabla llamada tabla de procesos, con una entrada por cada proceso. Cada entrada almacena información sobre el estado del proceso, el reparto de memoria, la situación de sus archivos abiertos, su información de contabilidad y planificación y todo lo que un proceso debe guardar para poder pasar de activo a inactivo (o en reposo) a los efectos de poder reiniciarlo como si nunca se hubiera detenido.

3.3-Hilo de ejecución

Hasta ahora se ha desarrollado el concepto de procesos de un solo hilo, es decir, procesos que solo ejecutan un solo programa. Algunos sistemas operativos permiten la ejecución de varios programas en un mismo proceso. A cada programa de un mismo proceso, se lo denomina hilo. Los hilos de un mismo proceso comparten los recursos del proceso, básicamente, la porción de memoria para datos y el estado de la lectura de archivos. Cuando un hilo modifica un dato en la memoria, los otros hilos acceden a ese dato modificado inmediatamente. El proceso sigue en ejecución mientras al menos uno de sus hilos de ejecución siga activo. Cuando el proceso finaliza, todos sus hilos de ejecución también han terminado. Asimismo, en el momento en el que todos los hilos de ejecución finalizan, el proceso no existe más y todos sus recursos son liberados.

3.4-Interrución - IRQ

La interrupción (también conocida como interrupción de hardware o petición de interrupción) es una señal recibida por el procesador de un ordenador, indicando que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar un código específico para tratar esta situación. Es una suspensión temporal de la ejecución de un programa, para pasar a ejecutar una subrutina del servicio de interrupción (generalmente montado en un chip específico o en la BIOS). Luego de finalizada dicha subrutina, se reanuda la ejecución del programa.

Las interrupciones surgen de las necesidades que tienen los dispositivos periféricos de enviar información al procesador principal de un sistema de computación. La primera técnica que se empleó fue que el propio procesador se encargara de sondear (*polling*) los dispositivos cada cierto tiempo para averiguar si tenía pendiente alguna comunicación para él. Este método presentaba el inconveniente de ser muy ineficiente, ya que el procesador constantemente consumía tiempo en realizar todas las instrucciones de sondeo.

El mecanismo de interrupciones fue la solución que permitió al procesador desentenderse de esta problemática y delegar en el dispositivo la responsabilidad de comunicarse con el procesador cuando lo necesitara. El procesador, en este caso, no sondea ningún dispositivo, sino que queda a la espera de que estos le avisen (lo «interrumpan») cuando tengan algo que comunicarle (ya sea un evento, una transferencia de información, una condición de error, etcétera).

3.5-Multiprocesamiento

Este es otro importante concepto dentro de la administración de procesos. A diferencia de la multiprogramación, esta gestión se basa en la distribución de los procesos en múltiples CPU dentro de una misma computadora y, también, múltiples CPUs pueden ser utilizados para ejecutar múltiples hilos dentro de un único proceso (Ver Figura 2).

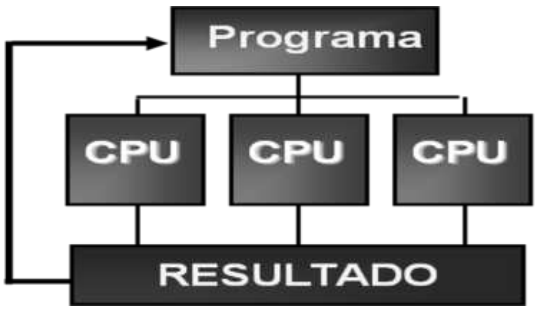


Figura 2

4-Entradas y salidas

En general, los SO tratan de estandarizar el manejo de las entradas y salidas. Pese a esto, existen algunos periféricos que requieren de una programación especial. La interfaz con periféricos debe ser simple de usar para que cualquier fabricante de ellos pueda adaptarse sencillamente. Los periféricos o dispositivos de entrada/salida se dividen en dos grandes grupos: por bloque y por caracter.

Los dispositivos por bloque intercambian con el SO bloques de información, que en general son de 512 bytes, pero también pueden ser mucho mayores. El típico periférico por bloque es el disco. La interfaz con estos dispositivos requiere que se intercambien direcciones (datos de bloque y pista) y el lote de bytes o datos.

El otro tipo de dispositivo es por caracter, que envían o reciben un flujo de caracteres que no son direccionables. Son ejemplos de estos dispositivos: las impresoras, el mouse, la interfaz de red, etcétera.

Esta estandarización permite que el SO independice su codificación del tipo de dispositivos de entrada/salida con el que esté interactuando.

4.1-Controladores de dispositivos

Normalmente la interfaz de entrada/salida está conformada por una parte electrónica y otra mecánica. Por ejemplo, la instalación de un escáner requiere que se instale en la computadora una placa o circuito electrónico que mediante un cable

se conecta a la parte mecánica. Estos circuitos electrónicos, también llamados controladores, muchas veces soportan más de un dispositivo mecánico de iguales características. Es conveniente que los fabricantes de los controladores y de la parte mecánica (que pueden ser los mismos o no) realicen los diseños de la interfaz bajo algún estándar, como ANSI, ISO, IEEE u otras. Esto facilita mucho la diversidad tecnológica y la posibilidad de independencia de proveedores.

Es importante comprender que el SO se relaciona con el controlador y no con el dispositivo mecánico. Los controladores tiene algunos cuantos registros que se utilizan para comunicarse con el SO. El SO solo debe conocer qué datos debe colocar o recibirá dentro de esos registros. Siempre se estable un espacio de memoria de la computadora para cada controlador o canal de comunicación, mediante la utilización de una relación establecida por un número de puerto lógico.

Por último, vale mencionar que los controladores deben manejar las llamadas para interrupción, como ya se comentó anteriormente.

5-Administración de memoria

El módulo del SO que administra la memoria se denomina administrador de memoria. Este se mantiene al tanto de qué partes de la memoria están en uso y cuáles no lo están. Asigna memoria a los procesos cuando estos la necesitan. Además, controla el intercambio de datos entre la memoria principal y la unidad de disco.

5.1-Monoprogramación

Se entiende por monoprogramación cuando la computadora solamente puede realizar un trabajo a la vez. El administrador de memoria solo acepta alojar en la memoria RAM al SO y un solo programa (Ver Figura 3).

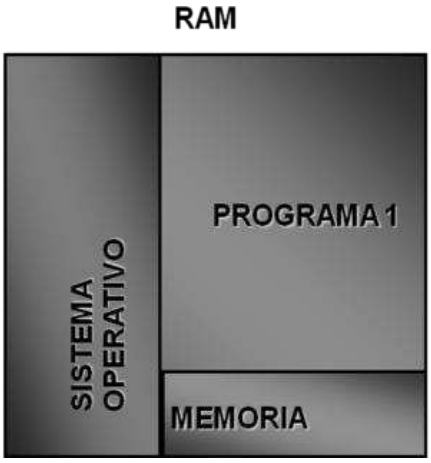


Figura 3

5.2-Multiprogramación

La multiprogramación, como ya se dijo, permite ejecutar varios procesos en una sola CPU, alternando la ejecución de estos a través de controles de tiempo, asignando límites temporales de ejecución a cada proceso y alternándolos a medida que este límite es alcanzado. Para lo cual el administrador de memoria destina un espacio de memoria a cada uno de los programas (Ver Figura 4).

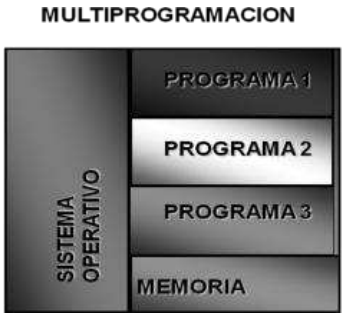


Figura 4

La figura anterior muestra cómo conceptualmente se subdivide la memoria. Pero lo que en realidad hace el SO es utilizar una técnica que se denomina «paginación». Esta técnica consiste en dividir los programas en pequeñas partes o páginas. Del mismo modo, la memoria es dividida en trozos, del mismo tamaño que las páginas de los programas ocuparán. A estos espacios de memoria, se los denominan marcos de página. En un momento cualquiera, la memoria se encuentra ocupada con páginas de diferentes programas, mientras que algunos marcos están disponibles para su uso. El SO mantiene una lista de estos últimos marcos y una tabla por cada programa, donde consta en qué marco se encuentra cada página del programa. De esta forma, las páginas de un proceso pueden no estar contiguamente ubicadas en la memoria y pueden intercalarse con las páginas de otros procesos.

El único inconveniente es que todas las páginas de un proceso deben estar en memoria para que se puedan ejecutar. Esto hace que si los programas son de tamaño considerable, no puedan cargarse muchos a la vez, disminuyendo el grado de multiprogramación del sistema. Para evitar esto, se permitirá que algunas páginas del proceso sean guardadas en un espacio de intercambio en la memoria secundaria (disco) mientras no se necesiten.

5.3-Almacenamiento virtual

Cuando la paginación se utiliza junto con memoria virtual, el SO mantiene además el conocimiento sobre qué páginas están en memoria principal y cuáles no, usando la tabla de paginación. Si una página buscada está marcada como no disponible (tal vez, porque no está presente en la memoria principal, pero sí en el área de intercambio), cuando la CPU intenta referenciar una dirección de memoria en esa página, la unidad de manejo de memoria responde levantando una excepción (comúnmente llamada fallo de página). Si la página se encuentra en el espacio de intercambio, el sistema operativo invocará una operación llamada intercambio de página, para traer a memoria principal la página requerida (Ver Figura 5).

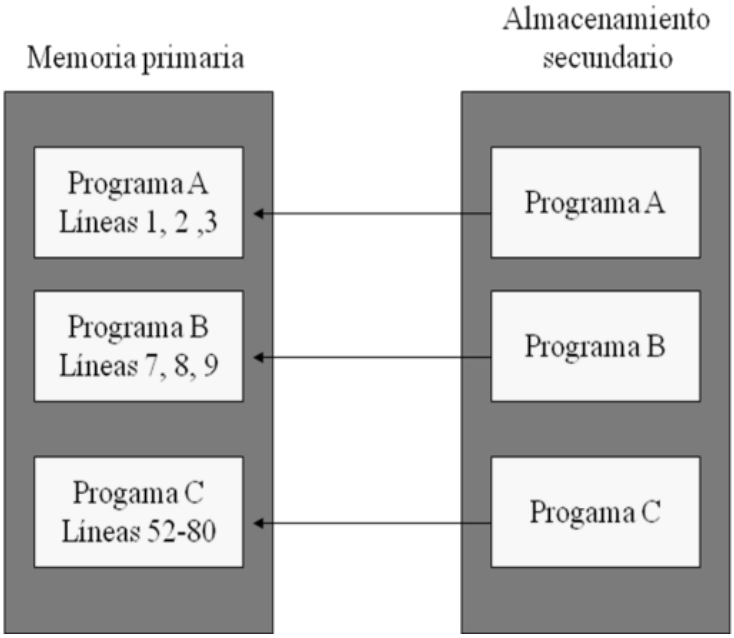


Figura 5

6-Archivos

Junto a los procesos, el otro gran componente de un SO es el sistema de archivos. Todo proceso requiere para su ejecución que los datos que va a utilizar se encuentren físicamente en algún sitio. Además, debe permitir almacenar una gran cantidad de información que debe sobrevivir a la terminación de un proceso y que, a su vez, debe poder ser accedida por varios procesos simultáneamente.

Lo que hace el sistema de archivos, precisamente, es permitir que esta información se pueda organizar de una manera lógica y sencilla. Todo SO debe brindar las

herramientas (en forma de llamadas al sistema o instrucciones) para que el sistema de archivos sea funcional.

Entre las principales, se pueden encontrar:

a) Creación.
b) Apertura y cierre.
c) Lectura y escritura.

Además, es fundamental que el SO pueda gestionar la seguridad de la información contenida en los archivos.

Un archivo realmente es una colección de bytes relacionados bajo un único nombre. A su vez, los archivos se encuentran organizados bajo una estructura que los relaciona lógicamente, esta estructura se denomina directorio (algunos sistemas los denominan también como carpetas). También, los directorios tienen su organización. Se puede observar que los archivos están contenidos dentro de los directorios y estos insertos en una estructura jerárquica, la cual tiene un inicio en el directorio raíz (algunos sistemas operativos como los derivados de MS DOS (este incluido) tienen un directorio raíz por cada unidad de disco lógico). Además, los directorios deben tener su gestión de seguridad incorporada dentro del SO (Ver Figura 6).

Un directorio no es más que un archivo que contiene nombres de archivos o de otros directorios y sus ubicaciones físicas.

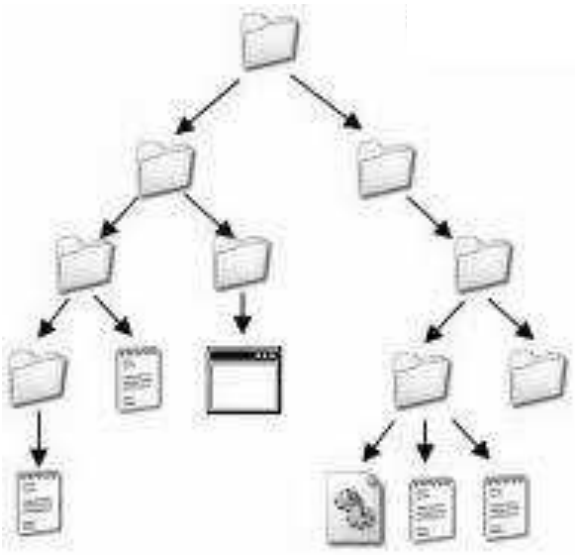


Figura 6

Con esta metodología, los usuarios pueden crear tantos directorios como requieran para luego agrupar dentro de ellos sus archivos.

Dado que los archivos se encuentran dentro de los directorios, se requiere de una regla para establecer los nombres de los archivos. Hay dos métodos diferentes en uso: direcciones absolutas y direcciones relativas. Con el primero, se asigna a cada archivo un nombre de ruta absoluto que consiste en la ruta desde el directorio raíz hasta el archivo. Por ejemplo, en Unix la ruta /datos/facturas/mayoristas.dat indica que el directorio raíz contiene un subdirectorio datos, que a su vez contiene un subdirectorio facturas y luego, dentro de este último subdirectorio un archivo que se denomina mayoristas.dat. Los nombres de ruta absolutos siempre parten del directorio raíz. En MS-DOS, el separador es "\". En este sistema, se debe especificar la unidad de disco dentro del directorio absoluto. El nombre relativo es el nombre de ruta relativo, que está en relación al directorio de trabajo (también llamado directorio actual). Un usuario puede especificar la ruta a partir del directorio actual. Por ejemplo, si el directorio de trabajo actual es /datos/facturas/, entonces se podrá hacer directa referencia al archivo de ese directorio (mayoristas.dat) sin necesidad de especificar la ruta. En la mayor parte de los SOs, cada proceso tiene su propio directorio de trabajo. Si la ruta comienza directamente en el nombre de un directorio, es una ruta relativa a partir del directorio de trabajo. Por ejemplo, si el directorio de trabajo es /datos/facturas/ y se invoca una dirección, como octubre/primeraquincena/mayoristas.dat, esta dirección absoluta sería /datos/facturas/octubre/primeraquincena/mayoristas.dat.

6.1-Partición

Una partición de disco es el nombre genérico que recibe cada división presente en una sola unidad física de almacenamiento de datos. Toda partición tiene su propio sistema de archivos que se establece durante su formateo. Generalmente, casi cualquier sistema operativo interpreta, utiliza y manipula cada partición como un disco físico independiente, a pesar de que dichas particiones estén en un solo disco físico. El objetivo básico de una partición de disco es permitir la posibilidad de alojar más de un SO en un disco duro. Las particiones de un disco están definidas en el último sector del disco. Actualmente, para definir la partición de un disco, se está utilizando un estándar denominado «Tabla de partición GUID (GPT)». Es parte del estándar *Extensible Firmware Interface* (EFI) propuesto por Intel.

6.2-Técnica de gestión de archivos

El aspecto fundamental de la gestión de archivos se refiere a cómo los nombres de archivos se relacionan a los lugares físicos de un disco. Esta relación se establece utilizando diferentes métodos a los que se denomina: Sistema de Archivos. Cada SO

utiliza un determinado Sistema de Archivos, algunos de los cuales se verán a continuación.

6.2.1-Asignación contigua

En este sistema, un archivo está integrado por bloques contiguos de datos en el disco. Así, en un disco con bloques de 1K, un archivo de 100K recibirá 100 bloques contiguos. Para encontrar un archivo con este método, hay que referenciarse a la ubicación del primer bloque del archivo. Se sabe que el resto del archivo son los bloques contiguos. Una gran ventaja de este método es que con una sola operación de lectura se puede leer todo el archivo. Pero la desventaja radica en que se debe conocer, de antemano, la longitud total del archivo. Esta información es necesaria para que el SO pueda reservar el espacio necesario.

El SO guarda en alguna tabla del sistema la relación entre el nombre del archivo y el primer bloque del mismo.

6.2.2-Asignación por lista enlazada

Este método consiste en guardar una lista enlazada de bloques de disco. La primera palabra de cada bloque se emplea como apuntador del siguiente. El resto del bloque se destina a datos (Ver Figura 7).

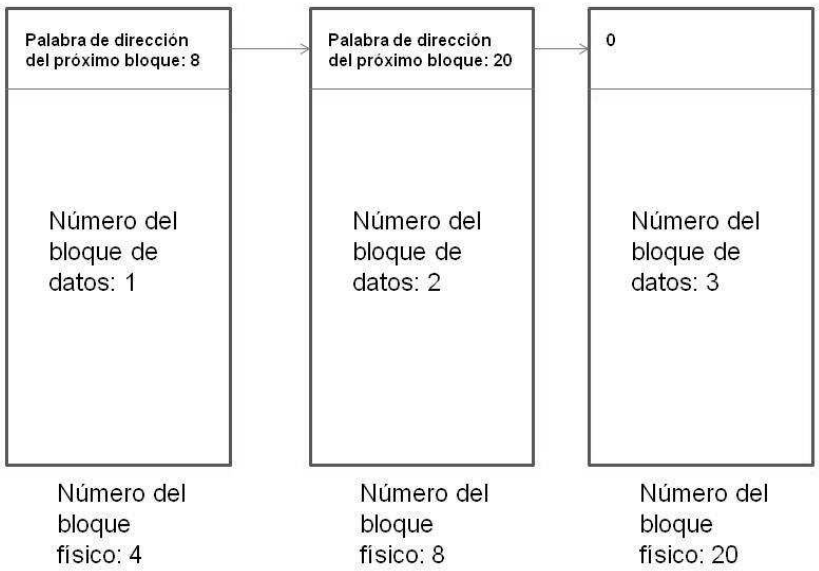


Figura 7

Este método evita tener que saber de antemano la longitud total del archivo. Además, basta con que en la entrada del directorio se almacene la dirección en disco del primer bloque para conseguir la totalidad del archivo siguiendo los enlaces. La lectura secuencial de este método es simple, pero el acceso directo se hace más complejo, porque siempre se debe ir al primer bloque para luego encontrar los siguientes.

Al igual que el método anterior, el SO guarda una tabla que direcciona a cada archivo al primer bloque del mismo.

6.2.3-Asignación por lista enlazada con uso de un índice

En los métodos anteriores, el SO debe recorrer los bloques y leer la palabra y sus respectivos apuntadores. Este método hace lenta la localización de los archivos. En este método, se emplea una tabla o índice en la memoria. Esta tabla contiene una relación entre todos los bloques del disco y si en ellos comienza un archivo, indican qué archivo comienza. Esto facilita la ubicación de los archivos en el disco (Ver Figura 8). El SO guarda esta tabla en algún lugar específico del disco.

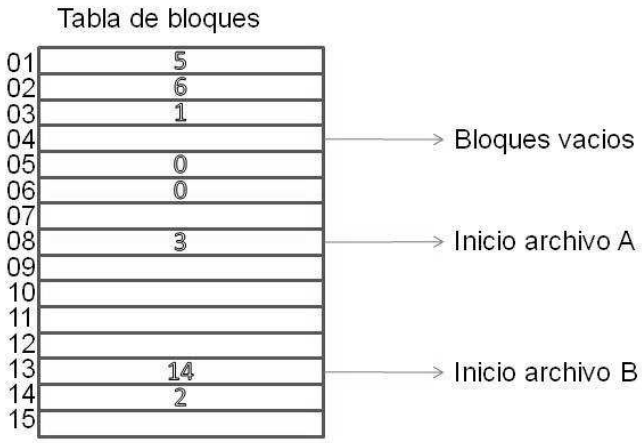


Figura 8

En este ejemplo, el archivo A comienza en el bloque 08, luego continua en el 3, luego pasa al 1, luego al 5 donde finaliza.

El archivo B comienza en el bloque 13, luego continua en el 14, luego pasa al 2, luego al 6 donde finaliza.

6.2.4-Nodos-i

Un inodo o nodo índice es una estructura de datos que relaciona un grupo de bloques de un dispositivo con un determinado nombre de archivo. Internamente, el núcleo del SO no distingue sus archivos por su nombre, sino por un número de inodo; de esta forma, el fichero con número de inodo 23421 será el mismo tanto si se denomina *passwd* como si se denomina *fichero*.

El SO crea una tabla de inodos en la que almacena, por cada inodo, la siguiente información (Ver Figura 9):

- Tipo de archivo: normal, directorio, pipe, especial.
- Identificación del propietario.
- Número de enlaces: esto es cuando el SO permite que el archivo esté presente en más de un directorio.
- Tamaño del archivo.
- Fecha de último acceso.
- Fecha de última modificación.
- Un *array* (vector) de 13 posiciones que se utiliza de la siguiente forma:

  El primer bloque del archivo se asigna a la primera posición del *array*. El segundo bloque que se asigne al archivo se coloca en la segunda posición del *array*. Y así sucesivamente hasta la posición 10. Si el SO requiere asignar un nuevo bloque al archivo, en la entrada 11 asigna la dirección de un bloque (que se denomina bloque indirecto sencillo) que se utilizará para asignar bloques, es como una extensión del *array* de la tabla de inodos. En este bloque indirecto sencillo, se establecerán los nuevos bloques hasta que su capacidad lo posibilite. Si fuese necesario continuar asignando bloques, el SO en la entrada 12 del *array* ingresa la dirección de un bloque (al que se denomina doble indirecto). Este bloque doble indirecto es un contenedor de direcciones de bloques indirectos sencillos, los cuales contendrán direcciones de bloques de datos. Cuando este último bloque indirecto sencillo esté colmado, en el bloque doble indirecto se asignará un nuevo bloque indirecto sencillo y así sucesivamente hasta colmar el bloque doble indirecto. Si el archivo requiere aún más bloques, en la posición 13 del *array* de inodos se asigna la dirección de un nuevo bloque de direcciones de bloques, al que se denomina triple indirecto. Dentro del bloque triple indirecto, se asignan bloque de direcciones de bloques doble

indirecto al que se le pueden asignar direcciones de bloques simples indirectos. Y así sucesivamente hasta que se colme el bloque de asignación de bloques triples indirectos. Esta técnica permite que si el bloque se identifica con un número entero de cuatro bytes y el tamaño del bloque es de 1kB, entonces un bloque puede contener 256 números de bloques. Por lo tanto, el total de información que podría contener un archivo sería igual a 10 + 256 + 256 x 256 + 256 x 256 x 256 = 16.843.018 KB.

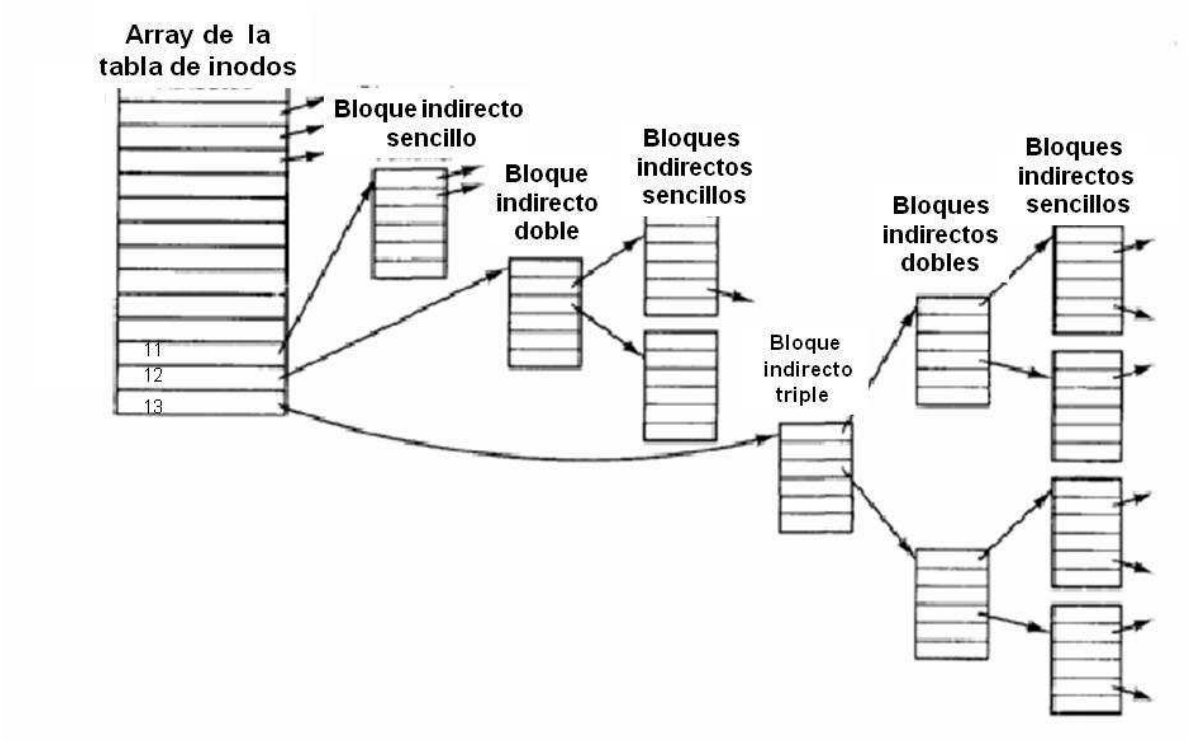


Figura 9

6.3-Organización de los archivos

La organización de los archivos es un aspecto relacionado con la forma en que la información se almacena dentro de los archivos. La organización de los archivos es relativamente independiente de la forma en que el SO gestiona los archivos en los soportes de información. Este aspecto estará más desarrollado cuando se describan los sistemas de gestión de datos. La organización de los archivos depende fundamentalmente del tipo de aplicación que se desarrolle. Los programas gestores de la organización de archivos deberán arreglárselas con las funcionalidades que le brinde el SO para gestionar archivos, ya sea que se trate de asignación contigua,

asignación por lista enlazada, asignación por lista enlazada con uso de un índice o nodos-i.

6.3.1-Secuencia de bytes

Una forma de organizar los archivos es como una secuencia de bytes. Esta visión ofrece el máximo de flexibilidad. Las aplicaciones de los usuarios pueden arrojar cualquier información que deseen a los archivos. Cualquier significado que se le quiera dar a los datos dependerá de la aplicación del usuario.

6.3.2-Secuencia de registros de longitud fija

Otra forma de organizar un archivo es la de secuencia de registros de longitud fija (Ver Figura 10). Este concepto se apoya en la suposición de que una operación de lectura devuelve un registro y que la operación de escritura sobrescribe o anexa un registro. Cuando en realidad, como se vio anteriormente, la lectura y grabación de los discos se realiza por bloques y no por registros. Muchos viejos SOs mantuvieron durante mucho tiempo la tradición de leer o grabar registros de 80 caracteres. Esta metodología derivó de cuando se utilizaba la tarjeta perforada de 80 columnas, base de la tecnología de la información durante varios años.

También, se dice que los registros están formados por un campo o un conjunto de campo. El campo es la primera unidad simbólica con sentido para una aplicación de usuario. Es un concepto lógico que manejan los programas aplicativos. En los SOs de hoy solo se manejan conceptos como discos, cilindros, pistas o bloques. El concepto de registro y campo pertenece al nivel de aplicación usuaria o del sistema de gestión de datos.

Esta organización es denominada secuencial, ya que la búsqueda de los registros debe hacerse en forma secuencial.

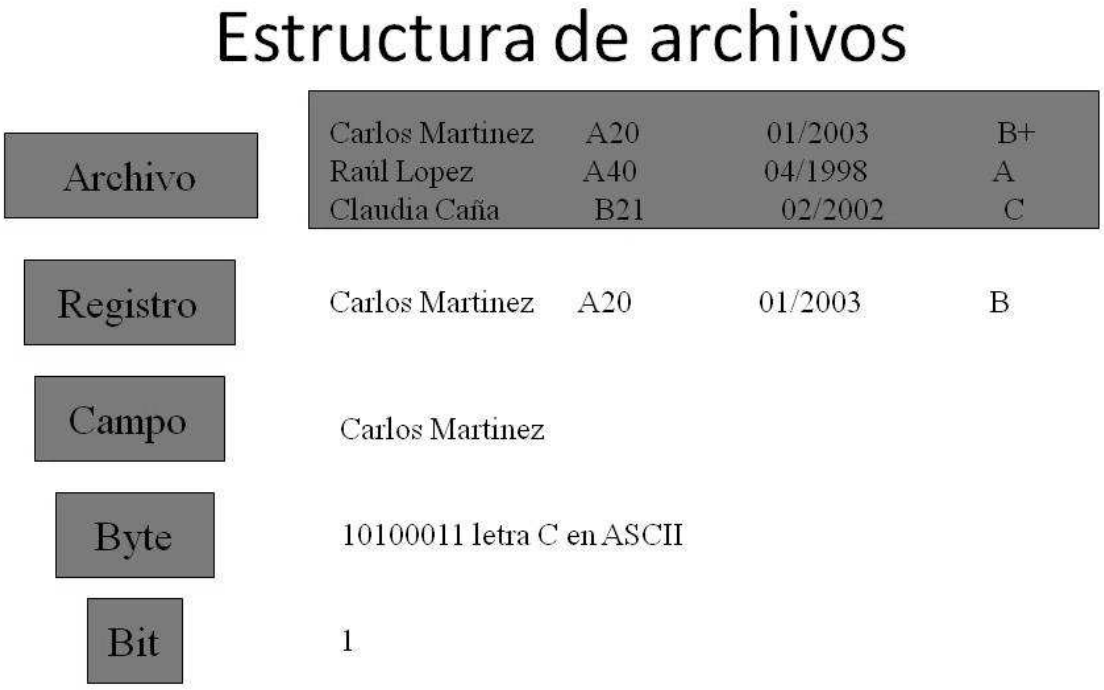


Figura 10

6.3.3-Secuencial indexada

Y una tercera forma de organizar los archivos consiste en un árbol de registros de igual o diferente longitud. Cada uno de ellos contiene una clave en una posición fija dentro del registro. Esto es un identificador inequívoco de cada registro, que puede estar formada por uno o un conjunto de campos. Por ejemplo: en un archivo de personal, podría ser el campo calve el número de legajo o el número de DNI.

El árbol está ordenado según la clave, a fin de poder buscar rápidamente un registro en particular. A esta organización se la suele llamar como secuencial indexada, ya que los registros se pueden buscar por la clave o en forma secuencial (Ver Figura 11).

Un archivo secuencial indexado contiene un área de índices, un área de datos y un área de excedentes (*overflow*). Es necesario que los registros contengan un campo clave para identificarlos y que estén almacenados en un soporte de acceso directo. El área de índices agiliza la búsqueda dentro del archivo. El índice puede almacenarse en un lugar concreto del archivos (determinados bloques) o en un archivo distinto del que contiene los datos. El área de excedente se utiliza para agregar nuevos

registros. Cuando esta área de excedente se agranda demasiado, es necesario realizar una tarea de mantenimiento (reorganización) para estructurar nuevamente los índices y los respectivos datos. Para evitar grandes espacios excedentes, cuando se crean los archivos, se dejan espacios libres tanto en el contenedor de los índices como en el contenedor de los datos. En estos espacios libres, se agregan las actualizaciones.

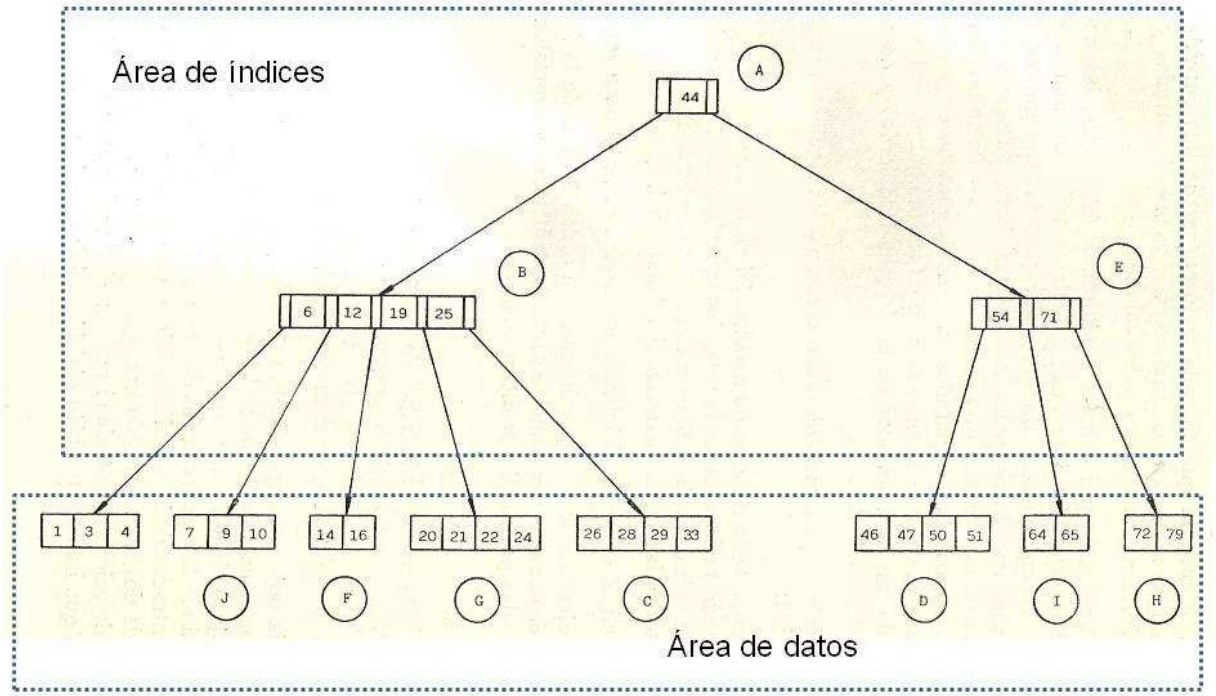


Figura 11

7-Llamadas al SO

Las llamadas al sistema son unas instrucciones especiales que utiliza el SO para su comunicación con los programas y de estos hacia el sistema. Están estrechamente relacionadas con el SO que se esté utilizando. A continuación, se agregan algunas llamadas, a modo de ejemplo, del SO UNIX:

Llamada	Descripción
Administración de procesos	
`pid = fork ( )`	Crea un proceso hijo y retorna su identificador de proceso
`s =kill (pid, sig)`	Envía una señal *sig* al proceso identificado por *pid*.
Archivos y directorios	
`fd = open (file, how)`	Abre un archivo *file*, para lectura o escritura (dependiendo de *how*), retorna un descriptor de archivo *fd*
`s = close (fd)`	Cierra el archivo descrito por *fd* si este se encuentra abierto
`n = read (fd, buffer, nbytes)`	Lee *nbytes* de datos del archivo descrito por *fd* y los coloca en el almacen *buffer*
`n = write (fd, buffer, nbytes)`	Escribe *nbytes* de datos al archivo descrito por *fd*, tomados desde el almacen *buffer*
`s = mkdir (name, mode)`	Crea un directorio con nombre *name*
`s = rmdir (name)`	Elimina el direcorio *name* siempre y cuando está vacío
`s = chdir (name)`	Cambia el directorio de trabajo a *name*

Estos son pequeños programas que hacen de interfaz entre el SO y la aplicación. Existe una norma internacional de llamadas al SO que se denomina Poxis (acrónimo del inglés *Portable Operating System Interface*). Son una familia de estándar de llamadas al sistema operativo definido por el IEEE y especificados formalmente en el IEEE 1003. Persiguen generalizar las interfaces de los sistemas operativos para que una misma aplicación pueda ejecutarse en distintas plataformas. Estos estándares surgieron de un proyecto de normalización de las API (acrónimo del inglés *Application Programming Interface*) describen un conjunto de interfaces de aplicación adaptables a una gran variedad de implementaciones de SOs.

Desde que la IEEE empezó a cobrar altos precios por la documentación de POSIX y se ha negado a publicar los estándares, ha aumentado el uso del modelo *Single Unix Specification*. Este modelo es abierto, acepta entradas de todo el mundo y está libremente disponible en Internet. Fue creado por The Open Group.

8-Seguridad

Los temas relacionados con la seguridad son un aspecto fundamental de la tecnología en los SOs. Hoy es uno de los desafíos principales del managment de la tecnología de la información en las organizaciones. Este desafío consiste en un buen equilibrio entre las normas de seguridad implementadas y el mantenimiento de un sistema eficiente.

A pesar de que es un tema crítico en todo SO, la seguridad de estos ha sido burlada en montones de oportunidades, incluso, existen casos muy famosos. Y, desde luego, seguirán siendo penetrados a pesar de los esfuerzos que se realicen. La seguridad absoluta no existe. Los SOs son programas muy complejos, que están en constate actualización y donde se integran productos de software y hardware de múltiples proveedores que también están en permanente actualización. No existe la posibilidad comercial de probar un SO lo suficiente antes de su liberación. Esta tarea podría demorar años y la competencia no lo permite.

El sistema de protección primario consiste en que el SO debe conocer a la persona que está ingresado al sistema. Para esto, se utiliza algún mecanismo de autenticación de usuario que el sistema solo conoce o posee. El método más utilizado es el de ingresar una clave que el SO inmediatamente cifra y busca en un archivo que contiene las claves de los usuarios autorizados también cifradas. Si la encuentra en el archivo, autoriza su ingreso. Si las claves no guardan cierta lógica de seguridad y se deja que el usuario las cree arbitrariamente, suelen ser muy fácilmente vulnerables. Las claves generadas por los usuarios responden a patrones nemotécnicos, muy estudiados por los especialistas en intrusión y cargados en bibliotecas de claves (hallables en Internet) que se utilizan para intentos de penetración, en general, exitosos.

Existe una gran cantidad de métodos de autenticación y aspectos de seguridad a tener en cuenta, pero que pertenecen a un capítulo aparte de lo que es el tema de los SOs, ya que abarca no solo aspectos del SO, sino que se requiere ver el software específico, el hardware específico y hasta temas relacionados con lo que se denomina ingeniería social (que es la técnica de obtener la información mediante la manipulación y persuasión de los usuarios legítimos de la información) entre otros.

Capítulo III
Sistema de administración de base de datos

1-DBMS

Los sistemas de administración de bases de datos (en inglés, *Data Base Management System*, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre el SO y las aplicaciones para la gestión de los datos (Ver Figura 12). Como se ha visto, el SO, básicamente, identifica los archivos por un nombre o un número y les asigna bloques de datos en los discos. Con anterioridad a los DBMS, las aplicaciones debían gestionar los datos dentro de los archivos con las pocas herramientas que los SOs brindaban. Los SOs trataban de no complejizar mucho sus gestión de archivos con el objetivo de estandarizar las llamadas, y que las aplicaciones pudieran construirse pensadas en función de ser ejecutadas en distintos SOs. Esto dio oportunidad a la aparición de los DBMS que solucionan la adaptación de las aplicaciones a los distintos SOs y les brindan una potente herramienta de gestión de datos.

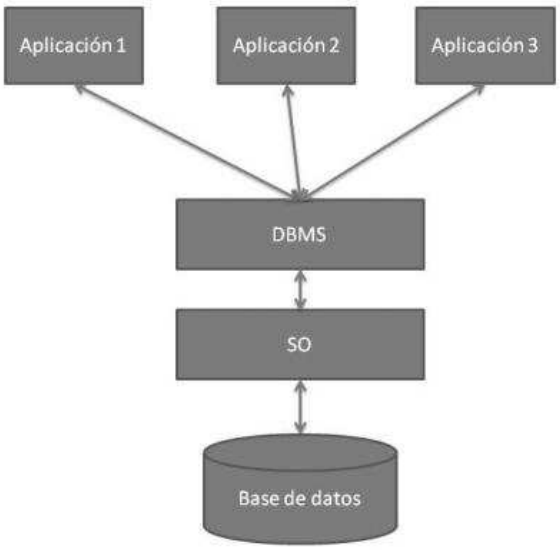


Figura 12

Las funcionalidades que brindan los DBMS son:

1. Mantener estructuras de datos complejas.
2. Proteger los programas frente a cambios en los archivos.
3. Reconstrucción de archivos.
4. Brindar seguridad a los datos.
5. Controlar los accesos simultáneos.

6. Se adaptan a los distintos SOs.
7. Efectividad en la búsqueda de información.

1.1-Mantener estructuras de datos complejas

El diccionario de datos es el reservorio de información sobre los datos de la organización (Ver Figura 13). Son datos sobre los datos a los que se denominan metadatos. Contiene la sintaxis y la semántica de los datos. Se incluye el nombre, la descripción, el alias, valores por defecto, valores máximos y mínimos, listas de opciones, máscaras de entrada, formatos, reglas de validación, etiquetas, descripción del significado, etcétera. También, brindan información de todos los elementos que forman parte del flujo de datos en todo el sistema (almacenes y procesos) y sus transformaciones.

Si los analistas desean conocer cuántos caracteres abarca un determinado dato o qué otros nombres recibe en distintas partes del sistema, o dónde se utiliza, etcétera, encontrarán las respuestas en un diccionario de datos desarrollado en forma apropiada.

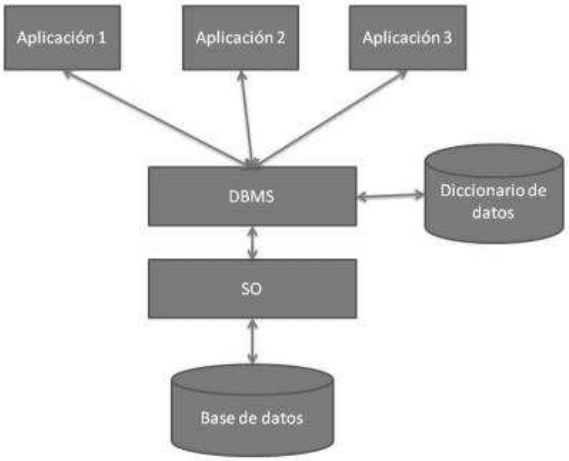


Figura 13

1.2-Proteger los programas frente a cambios en los archivos

Anteriormente a los DBMS, cada programa que se construía debía contener el diseño de los archivos que utilizaría dentro del conjunto de instrucciones y entenderse directamente con el SO (Ver Figura 14).

Ejemplo:

Un programa que utilizaba el archivo maestro de cuentas contables y un archivo de transacciones contables debía indicar dentro del conjunto de instrucciones información sobre: cómo se llaman los archivos, dónde se encuentran, qué campos contienen (con sus respectivos nombres), qué longitud y qué característica tienen cada campo. Esta información servía a la aplicación para identificar cada registro dentro de un archivo y procesar los datos correspondientes. Igual tarea debía hacer cada programa que maneje estos mismos archivos. Por lo tanto, en una aplicación de múltiples programas, existía una repetición enorme de instrucciones para simplemente definir los archivos. Además, cada vez que un archivo quería ser modificado, no quedaba más remedio que ingresar a cada programa y efectuar los cambios a la estructura del archivo modificado. Tarea tediosa y riesgosa.

Con los DBMS, esta tarea se simplifica mucho, ya que la definición de los archivos (tablas en lenguaje de DBMS) se encuentra en el diccionario de datos. Las aplicaciones desconocen la ubicación del archivo, su longitud y la totalidad de los campos. Solamente debe conocer los nombres de los campos que utilice y de qué tabla obtenerlos. Del resto se encarga el DBMS (Ver Figura 15). Si se requiere cambiar la longitud de un registro de un archivo, con cambiar la información en el diccionario del DBMS será suficiente.

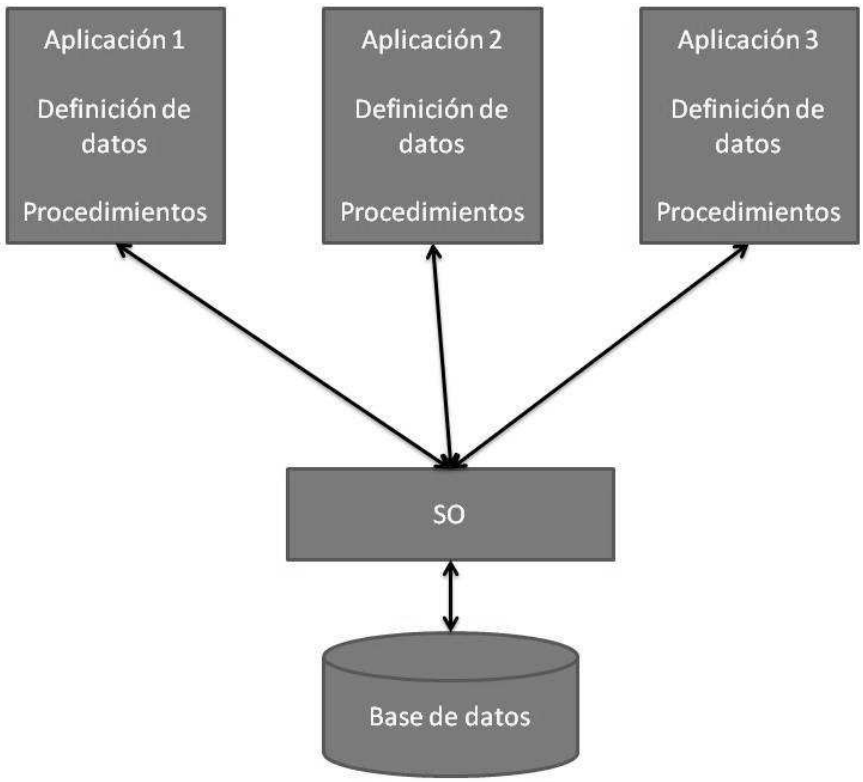


Figura 14

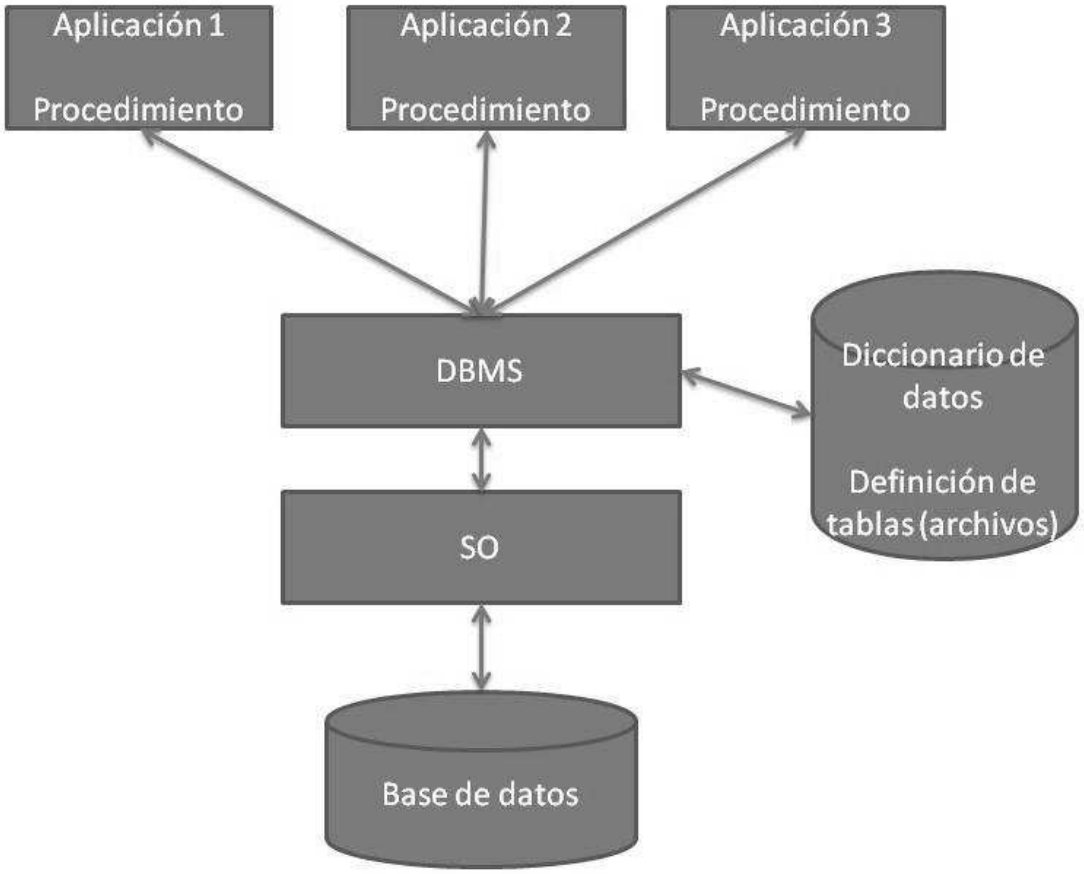


Figura 15

1.3- Reconstrucción de archivos

Un DBMS mantiene un registro de todos los cambios realizados sobre los datos en un período de tiempo. Esto permite la reconstrucción de los datos si se produce un fallo en el sistema, como un corte no previsto de la corriente eléctrica. Cuando se define una transacción, se agrupan eventos (básicamente de actualización de tablas), que deben terminarse satisfactoriamente. Todos los eventos realizados por la transacción deben tener éxito, de lo contrario la transacción quedaría trunca (incompleta), creando una inconsistencia de los datos. Si ocurriera un fallo y la transacción quedase trunca, el DBMS debe volver la situación al comienzo de la transacción, como si la misma no hubiera existido. El objetivo del control por transacciones es llevar la base de datos de un estado consistente a otro estado consistente.

El ejemplo de esto es: una transacción de facturación (simplificando el proceso) normalmente actualiza una tabla de transacciones de ventas (de donde se emite el

subdiario de ventas) y una tabla de cuenta corriente por cliente (donde se llevan los débitos y créditos de cada cliente). Si la transacción registra la factura en la tabla de transacciones de ventas, pero por un fallo no logra hacerlo en la tabla de cuenta corriente, resultaría que la venta se efectuó, pero la deuda del cliente no estaría reflejada en su cuenta corriente. Por lo tanto, cuando el DBMS detecta que la transacción quedó trunca, debe eliminar el movimiento sobre la tabla de transacciones de ventas, de tal forma de volver la base de datos a un nuevo estado de consistencia, como si la transacción nunca hubiera existido. Para que esta funcionalidad opere correctamente, el programador de la aplicación debe incluir una instrucción de comienzo de transacción y una de finalización de transacción. Si esto no está programado de esta forma, el DBMS nunca se enterará de que existió una transacción que debería finalizar exitosa.

1.4- Brindar seguridad a los datos

Un DBMS debe permitir controlar el acceso al sistema de forma que solo los usuarios autorizados puedan manipular los datos de la base de datos como así también su estructura. Generalmente, existe una jerarquía de usuarios en cada base de datos, desde un administrador de la base de datos, que puede cambiarlo todo pasando por los usuarios con permisos para añadir o borrar datos, hasta los usuarios que solo pueden leer datos. El nivel de seguridad sobre los datos debe llegar a nivel tanto de tabla completa como de un dato específico de la tabla o de un determinado conjunto de registros. Esta funcionalidad la brindan con la técnica denominada de gestión de vistas. Es decir que los usuarios no acceden directamente a la tabla de un DBMS, sino que lo hacen por intermedio de una vista (que incluye determinadas columnas de la tabla, así como determinadas filas) que está autorizada a determinado/s usuario/s y con derechos diferenciados en cuanto a qué actividades puede realizar con la vista.

1.5-Controlar los accesos simultáneos

El acceso de las aplicaciones a los mismos datos en el mismo momento era una preocupación permanente en el desarrollo de las aplicaciones y difícil de resolver. Las aplicaciones debían, previamente, estipular una forma de gestionar los accesos concurrentes a los datos, cosa que en general requería de una solución bastante engorrosa en cantidad de instrucciones y por la creación de archivos especiales a tal efecto. Los DBMS brindan un conjunto de instrucciones muy simples para gestionar la concurrencia de los datos a las aplicaciones. En general, se manejan dos niveles de bloqueo básicos: 1) donde una aplicación puede bloquear datos o generar datos bloqueados, impidiendo que otras aplicaciones puedan verlos hasta que la aplicación que bloqueó los libere; 2) la aplicación bloquea datos existentes o que se estén generando, pero le permite a las otras aplicaciones que vean el dato simultáneamente.

1.6- Se adaptan a los distintos SOs

Uno de los aspectos más importantes de la utilización de los DBMS es la trasparencia que brinda a las aplicaciones respecto del SO y de una determinada marca de DBMS en concreto. La gran mayoría de los DBMS utilizan un lenguaje denominado SQL (que más adelante se detallará con mayor profundidad). Las aplicaciones utilizan este lenguaje (que además es un lenguaje estándar al que adhieren la casi totalidad de los DBMS del mercado) para comunicarse con los DBMS y gestionar los datos. Por lo tanto, si la aplicación se migra de un DBMS de una marca a otra, a la aplicación le debe resultar transparente. También, es trasparente para la aplicación, al menos en la gestión de los datos, el SO sobre el cual se ejecute la aplicación, ya que, precisamente, una de las virtudes del DBMS es servir de interfaz con el SO. La adaptación al SO será una tarea del DBMS, y cuanto más SOs soporte el específico DBMS, más independencia brindará a la aplicación.

1.7-Efectividad en la búsqueda de información

Los DBMS manejan múltiples y complejos algoritmos de búsqueda de información dentro de las entidades. Dentro de estas virtudes, existe la generación de claves a modo de ejecución, es decir, al momento que se está realizando la búsqueda. Y por otro lado, SQL brinda un lenguaje de consulta muy simple de aprender y muy poderoso en cuanto a las posibilidades que presenta.

2-Conceptos básicos sobre base de datos relacional

Una base de datos relacional está formada por un conjunto de tablas o entidades (que anteriormente y a nivel de SO se han venido denominado como archivos) que contienen la información necesaria por las funcionalidades de las aplicaciones programadas para una computadora. Por ejemplo:

MAESTRO-DE-CLIENTES				
nro-cliente	razon-social	CUIT	direccion	fecha-alta
25	cliente-a	xxx	xxxxx	xx/xx/xxxx
12	cliente-b	zzz	zzzzz	zz/zz/zzzz

El lector debe tener en claro que el nombre de la tabla MAESTRO-DE-CLIENTES, como los nombres de las columnas, no integran la tabla. Son conceptos definidos en el diccionario de datos del DBMS. Es decir que la tabla anterior está compuesta de dos filas únicamente.

Las filas de la tabla son elementos de la tabla, que se identifica por un dato inequívoco de cada fila en particular, al que se denomina clave de la tabla (en el

ejemplo anterior, es el **nro-cliente,** no puede haber dos filas con el mismo número de cliente). Y las columnas, que no son clave de la tabla, constituyen atributos de ese elemento (**razon-social**, **CUIT**, **direccion**, **fecha-alta**). A las columnas de las tablas, al igual que en los archivos, se las denominan campos.

Pueden existir tablas que requieran más de una columna o campo para su inequívoca identificación. Estas son claves compuestas, por ejemplo:

item	deposito	cantidad
a	1	20
a	2	56
b	1	11
b	2	45

En este caso, la clave de la tabla debe estar formada por los campos **item** y **deposito** en forma conjunta. El atributo **cantidad** corresponde al ítem en un determinado depósito. El lector puede observar que si se tomara el campo **item** únicamente como campo clave, este se repetiría en cada depósito que, al menos, tuviera existencias. El campo **deposito** se repite por cada ítem de existencia en el depósito. En cambio, la combinación de **item deposito** no se pude repetir porque un ítem, por lógica, puede tener un solo saldo en cada depósito.

2.1-Redundancia

La redundancia es la duplicidad de datos en múltiples tablas (entidades) y dentro de las mismas tablas.

2.1.1-Redundancia en múltiples tablas

Por ejemplo, podría ser cuando un DBMS contiene dos tablas como las siguientes:

MAESTRO-DE-CLIENTES				
nro-cliente	razon-social	CUIT	direccion	fecha-alta
25	cliente-a	xxx	xxxxx	xx/xx/xxxx
12	cliente-b	zzz	zzzzz	zz/zz/zzzz

FACTURACION						
nro-ctura	nro-cliente	razon-social	CUIT	total-bruto	IVA	total
10	25	cliente-a	xxx	100	21	121
11	12	cliente-b	zzz	200	42	242

En estos casos, se pude ver cómo la **razon-social** y el **CUIT** se repiten en las tablas de MAESTRO-DE-CLIENTES y en la de FACTURACION.

Esto no solamente genera un problema de duplicidad de información, sino que además admite la posible inconsistencia de la información. El lector puede ver que la tabla MAESTRO-DE-CLIENTES en el cliente 25 puede llamarse cliente-a; y en la tabla de FACTURACION, el cliente 25 podría llamarse cliente-x. Esto sería absolutamente inadmisible en una estructura de datos confiable, salvo que, como excepción, así se requiera para lograr una mejor performance de la aplicación.

2.1.2-Redundancia en una misma tabla

El siguiente es un ejemplo de redundancia en la misma tabla:

FACTURACION						
nro-factura	nro-cliente	razon-social	CUIT	articulo	descripcion	precio
10	25	cliente-a	xxx	102	Xxxxxx	1500
10	25	cliente-a	xxx	100	Zzzzzz	250
11	12	cliente-b	zzz	100	Zzzzzz	250

En esta tabla de FACTURACION, se puede ver cómo en la primera y segunda fila la **razon-social** se repite. Este sería el hipotético caso de una factura número 10 para el cliente 25, denominado cliente-a y a quien se le han vendido dos artículos, el 102 y el 100. Como se puede ver, también se duplica el **CUIT** y la **descripcion** del artículo. Como en el caso anterior, esto no solamente genera un problema de duplicidad de información, sino que además admite la posible inconsistencia de la información. En el ejemplo, podría ocurrir que en la primera fila el cliente 25 tenga como **razon-social** cliente-a; y que en la segunda fila, el mismo cliente se llame cliente-x. Un caso parecido podría ocurrir también con la **descripcion** del artículo 100 en las filas 2 y 3.

2.2-Requisitos que deben cumplir las tablas de una DBMS relacional

Las tablas de una base de datos relacional deben cumplir ciertos requisitos:

1. Todas las filas representan un elemento de la misma clase.
2. Cada columna tiene un nombre propio.
3. Todas las filas deben ser diferentes.

2.3-Normalización

En concreto, se pude decir que un DBMS relacional se caracteriza por agrupar los datos en tablas (que en el modelo de diseño se denominan entidades), que agrupan elementos de una misma clase. Para evitar redundancias e inconsistencias, la teoría de la base de datos relacional propone, en muchos casos, que las entidades se reorganicen en varias tablas y que estas se relacionen entre sí por intermedio de alguna columna o conjunto de columnas. La técnica para lograr este objetivo se denomina normalización de datos.

La normalización es un proceso que asegura que los datos sean agrupados dentro de las entidades en la forma más simple posible.

La normalización identifica los atributos que no dependen totalmente de la clave de una entidad, generando nuevas entidades desde estos atributos. Las formas normales definidas en la teoría de bases de datos relacionales representan una guía para el diseño de la base de datos relacional. Las reglas de normalización fueron diseñadas para prevenir la duplicidad de datos y las inconsistencias de los mismos.

2.3.1-Primera forma normal (1FN)

Se ocupa de la «forma» de un registro (fila de cada tabla), dada esta forma normal, todas las filas de una tabla deben contener el mismo número de campos. Esto excluye los registros con ocurrencias (repeticiones) variables. Por ejemplo:

FACTURACION

nro-factura	cliente	articulo1	cantidad1	precio1	articulo2	cantidad2	precio2	articulo3	cantidad3	precio3	importe-total
1	100	aa	5	30	bb	10	50	cc	20	10	850
2	101	cc	25	10	aa	40	30	2050			

La tabla anterior almacena las facturas 1 y 2 y en la misma fila, para cada factura, contiene los artículos facturados, que por lógica son variables en cada factura. En la

factura 1, se vendieron 3 artículos (aa, bb y cc); y en la factura 2, se vendieron 2 (cc y aa). Esto requiere que los campos **articulo**, **cantidad** y **precio** ocurran (se repitan) en la fila tantas veces como artículos se vendan en cada factura. Esta es una solución de registros (filas) de longitud variable que está totalmente excluida en el diseño de base de datos relacionales. Solamente se admiten registros (filas) de longitud fija.

2.3.2-Segunda forma normal (2FN)

Se ocupa de la relación entre los campos que forman parte de la clave y de los que no forman parte de la clave. Bajo la 2FN, todos los campos que no forman parte de la clave deben proveer un atributo acerca de la clave completa.

Considerando el siguiente registro de inventario:

item, deposito, cantidad, direccion-deposito

La clave está compuesta por **item** y **deposito**. El campo **direccion-deposito** es un atributo solo del depósito. Los problemas básicos con este diseño son:

- La dirección del depósito se repite en cada registro.
- Si la dirección del depósito es cambiada, cada registro que corresponde a ese depósito debe ser actualizado.
- A causa de la redundancia, los datos pueden ser inconsistentes. Varios registros pueden contener diferentes direcciones del depósito para un mismo depósito.
- Si en un determinado momento no existen ítems almacenados en el depósito, puede suceder que no queden registros en donde guardar la dirección del depósito.

Para satisfacer la 2FN, la tabla debería ser descompuesta en 2:

a) **item, deposito, cantidad.**

b) **deposito, direccion-deposito.**

El proceso por el cual un diseño de datos es cambiado tal cual se hizo, reemplazando registros no normalizados por registros normalizados, se conoce con el nombre de normalización.

EL diseño normalizado acrecienta la integridad de los datos, minimizando la redundancia y la inconsistencia con un posible mayor costo en la performance para las aplicaciones que requieren recuperar gran cantidad de datos. Por esto conviene realizar un diseño completamente normalizado, y luego, analizar los problemas de

performance en una segunda etapa, reduciendo, quizás, el nivel de normalización de algunos registros.

Considerando una aplicación que necesita obtener la dirección del depósito que almacena un determinado ítem. En la forma no normalizada, la aplicación deberá buscar un solo registro; en cambio, en la forma normalizada, la aplicación deberá buscar dos tipos de registros y conectar los pares apropiadamente.

2.3.3-Tercera forma normal (3FN)

La 3FN, al igual que la 2FN, se ocupa de la relación entre los campos que forman parte de la clave y de los que no forman parte de la clave. La 3FN es violada cuando un campo no clave es un atributo de otro campo no clave. Por ejemplo:

empleado, departamento, ubicacion-departamento

El campo clave es empleado, si cada departamento está ubicado en un lugar, entonces el campo **ubicacion** es un atributo del departamento.

Los problemas con este diseño son:

- La ubicación del departamento se repite en el registro de cada empleado asignado a ese departamento.
- Si la ubicación cambia, cada registro afectado debe ser actualizado.
- Los datos pueden ser inconsistentes.
- Si el departamento no tiene empleados, puede no haber registro para guardar la ubicación del departamento.

Para satisfacer la 3FN, la tabla debería ser descompuesta en 2:

a) **empleado, departamento.**

b) **departamento, ubicacion.**

Sintetizando, un registro está en 2FN y 3FN si cada campo que no es parte de la clave provee un atributo acerca de la clave completa.

2.3.4-Cuarta forma normal (4FN)

Los atributos de una entidad pueden ser de dos tipos:

a) Con valor único.
b) Con valores múltiples.

Suponiendo por ejemplo que la entidad **EMPLEADOS** tiene los atributos **nacionalidad, sexo, idioma y titulo-universitario**.

Los campos nacionalidad y **sexo** son atributos con valor único porque un empleado tiene una sola nacionalidad (se entiende que se hace referencia a la nacionalidad de origen y no a las nacionalidades por adopción) y un único sexo; en cambio, el campo **idioma** y el campo **titulo-universitario** son atributos con valores múltiples, porque un mismo empleado puede dominar varios idiomas y poseer varios títulos universitarios.

La primera, segunda y tercera forma normal se ocupan de atributos con valor único. La cuarta y quinta forma normal se ocupan de los atributos con valores múltiples y se refieren a claves compuestas. Estas formas normales ayudan a minimizar el número de campos incluidos en una clave compuesta.

Considerando los datos **empleados**, **habilidades** y **lenguajes**, donde un empleado puede tener varias habilidades y varios lenguajes y estas variables son independientes entre sí. Bajo la cuarta forma, estas dos relaciones no deberían tener representación en un registro simple como:

empleado, habilidad, lenguaje

En este caso, la clave debería estar formada por los 3 campos, debido a que un empleado puede tener más de una habilidad y más de un lenguaje.

Según la 4FN, este archivo debería abrirse en 2:

a) empleado, habilidad.

b) empleado, lenguaje.

Si los atributos **habilidades** y **lenguajes** tuvieran una relación de dependencia, la 4FN no sería adecuada para la solución.

El principal problema que se presenta con la violación de la 4FN es que genera incertidumbre en la política de mantenimiento. Se mostrarán tres soluciones que violan la 4FN para demostrar los inconvenientes que generan.

1) Una forma disyunta en donde el registro contiene una habilidad o un lenguaje, pero no ambos.
2)

empleado	habilidades	lenguaje
Pérez	Cocina	
Pérez	Taquigrafía	
Pérez		Francés
Pérez		Alemán
Pérez		Griego

Esto no es muy diferente que el mantener dos tipos de registros separados con la «contra» de generar ambigüedades por la presencia de campos en blanco. El blanco en el campo **habilidades** puede significar que la persona no tiene habilidades, que el campo no es aplicable a la persona o que el dato es desconocido o que, como en este caso, el dato puede encontrarse en otro registro.

2) Una mezcla al azar con 3 variantes en las cuales o se repiten atributos con la posible generación de inconsistencias o se dejan campos vacíos.

a) Cantidad mínima de registros con repeticiones.

empleado	habilidades	lenguaje
Pérez	Cocina	Francés
Pérez	Taquigrafía	Alemán
Pérez	Taquigrafía	Griego

b) Mínima cantidad de registros con valores nulos.

empleado	habilidades	lenguaje
Pérez	Cocina	Francés
Pérez	Taquigrafía	Alemán
Pérez		Griego

c) Sin restricciones.

empleado	habilidades	lenguaje
Pérez	Cocina	Francés
Pérez	Taquigrafía	
Pérez		Alemán
Pérez	Taquigrafía	Griego

3) Un producto cruzado en donde por cada empleado debe existir un registro por cada posible par de una de sus habilidades con uno de sus lenguajes.

empleado	habilidades	lenguaje
Pérez	Cocina	Francés
Pérez	Cocina	Alemán
Pérez	Cocina	Griego
Pérez	Taquigrafía	Francés
Pérez	Taquigrafía	Alemán
Pérez	Taquigrafía	Griego

Dependiendo de la solución adoptada, los problemas que se presentan pueden ser muy serios, tales como la inserción de pares de múltiples registros cuando se da de alta una nueva habilidad o un nuevo lenguaje para un mismo empleado.

2.3.5-Quinta forma normal (5FN)

La quinta forma normal se ocupa de casos en donde los datos pueden ser reconstruidos desde pequeñas piezas de Información que pueden ser mantenidas con muy baja redundancia. La segunda, tercera y cuarta forma normal también sirven a este propósito, pero la 5FN cubre casos no previstas por las otras.

Se ilustrará el concepto central con un ejemplo: si los vendedores representan compañías, las compañías hacen productos y los vendedores venden productos, se necesitará un registro indicando que producto vende cada vendedor y para qué compañía. Esta información podría guardarse en un registro de 3 campos:

vendedor	compania	producto
Campos	Ford	Automóvil
Campos	Renault	Camionetas

Esta es la forma necesaria en la generalidad de los casos. Se requiere de la combinación de los 3 campos para conocer cuáles combinaciones son válidas y cuáles no.

Ahora, suponiendo que existe la siguiente regla: Campos puede vender automóviles y camionetas de Ford y de Renault, y García solo puede vender automóviles Ford.

vendedor	compania	producto
Campos	Ford	Automóvil
Campos	Ford	Camionetas
Campos	Renault	Automóvil
Campos	Renault	Camionetas
García	Ford	Automóvil

En este caso, se podrá reconstruir toda la información desde registros normalizados de 3 tipos, cada uno debe contener 2 campos. El objetivo es minimizar la redundancia, la inconsistencia y la cantidad de registros.

1)

vendedor	compania
Campos	Ford
Campos	Renault
García	Ford

2)

vendedor	productos
Campos	Automóvil
Campos	Camionetas
García	Automóvil

3)

compania	productos
Ford	Automóvil
Ford	Camionetas
Renault	Automóvil
Renault	Camionetas

Estos 3 tipos de registro están en 5FN. En rigor, se dice que un tipo de registro está en 5FN si la información que contiene no puede ser reconstruida a partir de registros más pequeños.

La 5FN difiere de la 4FN en que contiene las restricciones en cuanto a las combinaciones que se pueden dar entre los atributos con valores múltiples. De no existir estas restricciones, los registros que se encuentren en 4FN también están en 5FN.

Una de las ventajas de la 5FN es que elimina ciertas redundancias. En la forma normalizada, el hecho de que Campos (ver el ejemplo anterior) está habilitado para vender Automóviles requiere un único registro; en cambio, en una forma no normalizada, requiere muchos registros.

Si bien la 5FN requiere de una mayor cantidad de tipos de registro, habitualmente produce una menor cantidad total de registros. Ello no se comprueba cuando la cantidad total de registros es pequeña (como en el caso del ejemplo ya visto). Esta ventaja aparece con toda claridad cuando una gran cantidad de atributos debe ser registrada. Por ejemplo, si se necesita agregar algún un nuevo vendedor que vende **x** productos de **z** compañía, y cada compañía hace cada uno de los **x** productos, se deberá agregar **x** + **z** nuevos registros de registros normalizados o **x** * **z** de registros no normalizados.

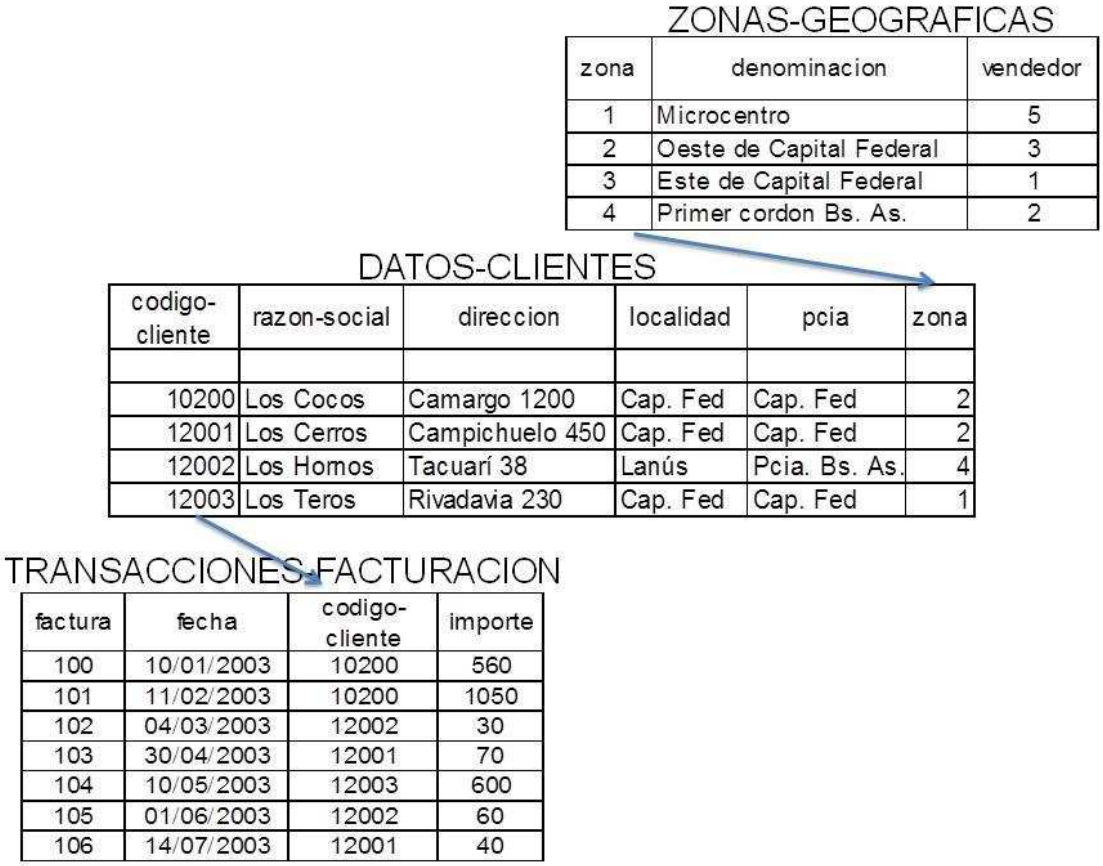
2.4-Integridad referencial

Las bases de datos relacionales están constituidas por relaciones entre tablas de doble entrada (filas y columnas) como ya se explicó (como se puede ver en el ejemplo de la figura 16). La tabla de **ZONAS-GEOGRAFICAS** está relacionada con la tabla de **DATOS-CLIENTES** por intermedio de la columna **zona**; la relación es de una fila en la tabla de **ZONAS-GEOGRAFICAS** a muchas posibles filas en la tabla de DATOS-CLIENTES con igual **zona**. La tabla de **DATOS-CLIENTES** está a su vez relacionada con la tabla de **TRANSACCIONES-FACTURACION**, por intermedio de la columna **codigo-cliente**, la relación es de una fila en la tabla de DATOS-CLIENTES a

muchas posibles en la tabla de **TRANSACCIONES-FACTURACION** con igual **codigo-cliente**. La tabla de **TRANSACCIONES-FACTURACION** contiene datos operacionales, mientras que las tablas de **DATOS-CLIENTES** y la de **ZONAS-GEOGRAFICAS** contienen datos referenciales. La columna **razon-social** de la tabla **DATOS-CLIENTES** es un dato referencial de la columna **codigo-cliente** de la tabla de **TRANSACCIONES-FACTURACION**; y la columna **denominacion** de la tabla **ZONAS-GEOGRAFICAS** es un dato referencial de la columna **zona** de la tabla **DATOS-CLIENTES**.

Los buenos DBMS almacenan todas las relaciones entre los datos en el diccionario de datos del sistema. Por intermedio de esta información, puede controlar que al ingresar una transacción, siguiendo el ejemplo, en la tabla **TRANSACCIONES-FACTURACION** se verifique que exista el código de cliente en la **tabla DATOS-CLIENTES**. De lo contrario, el DBMS estaría permitiendo una inconsistencia de la información en la base de datos. Esta forma de referenciar la información es una derivación de normalizar la información.

Además, las relaciones almacenadas en el DBMS suelen ser utilizadas para facilitar la generación de aplicaciones, como consultas, reportes, forms, etcétera.



ZONAS-GEOGRAFICAS

zona	denominacion	vendedor
1	Microcentro	5
2	Oeste de Capital Federal	3
3	Este de Capital Federal	1
4	Primer cordon Bs. As.	2

DATOS-CLIENTES

codigo-cliente	razon-social	direccion	localidad	pcia	zona
10200	Los Cocos	Camargo 1200	Cap. Fed	Cap. Fed	2
12001	Los Cerros	Campichuelo 450	Cap. Fed	Cap. Fed	2
12002	Los Hornos	Tacuarí 38	Lanús	Pcia. Bs. As.	4
12003	Los Teros	Rivadavia 230	Cap. Fed	Cap. Fed	1

TRANSACCIONES-FACTURACION

factura	fecha	codigo-cliente	importe
100	10/01/2003	10200	560
101	11/02/2003	10200	1050
102	04/03/2003	12002	30
103	30/04/2003	12001	70
104	10/05/2003	12003	600
105	01/06/2003	12002	60
106	14/07/2003	12001	40

Figura 16

3-SQL

SQL nació en 1975, en el marco de un proyecto de investigación de IBM denominado System/R, cuyo objetivo era demostrar la operabilidad de un DBMS relacional (luego se comentará el significado de relacional). EL DBMS relacional es un tipo especial y, largamente, el más difundido de los DBMS. El proyecto System/R incluía el desarrollo de un lenguaje de consulta para este tipo de DBMS. El lenguaje resultante se denominó SEQUEL, un acrónimo de *Structured English Query Language* (Lenguaje Estructurado de Consultas). Posteriormente, con la primera implementación experimental de este tipo de DBMS, se renombró bajo el acrónimo de SQL o *Structured Query Language* (Lenguaje Estructurado de Consultas). El proyecto, que había sido publicado en una revista científica de EE. UU., interesó a un grupo de ingenieros de Menlo Park, California, quienes en 1977 formaron una compañía llamada Relational Software Inc. para construir un DBMS relacional basado en SQL. El producto logrado se denominó Oracle y en 1979 comenzó a comercializarse adelantándose dos años a la finalización del trabajo de System/R de IBM. Relational Software Inc., hoy renombrada como Oracle Corporation, es actualmente la número uno en ventas de DBMS. SQL recibió el gran espaldarazo cuando en 1986 se publicó el estándar ANSI/ISO, ya no solo como un lenguaje de consulta, sino también de actualización de datos. De esta forma, los DBMS con lenguaje SQL comenzaron a adquirir una gran popularidad, a tal punto que prácticamente no existen sistemas de gestión de la información que no utilicen un DBMS relacional.

Con SQL se formulan operaciones relacionales, es decir, operaciones que permiten definir y manipular una base de datos relacional. Aun cuando se describe a SQL como un lenguaje de consulta, en realidad es mucho más que eso, ya que dispone de funciones para:

- Manipulación de datos.
- Definición de datos.
- Especificar restricciones de acceso y seguridad.
- Control de transacciones.
- Programático.

3.1-Manipulación de datos

Las instrucciones más utilizadas son:

SELECT: para recuperar datos de la base de datos.
INSERT: añade filas de datos a las tablas.
DELETE: suprime filas de las tablas.
UPDATE: modifica datos existentes en las tablas.

3.2-Definición de datos

CREATE TABLE: añade una tabla a la base de datos.
DROP TABLE: suprime una tabla a la base de datos.
ALTER TABLE: modifica la estructura de una tabla existente.
CREATE VIEW: añade una vista a la base de datos.
DROP VIEW: suprime una vista a la base de datos.
CREATE INDEX: construye un índice para una columna.
DROP INDEX: suprime un índice para una columna.
CREATE SYNONYM: crea un alias para una tabla.
DROP SYNONYM: suprime un alias para una tabla.
COMMENT: ingresa comentarios para una tabla.
LABEL: define el título de una columna.

3.3-Seguridad de datos

GRANT: concede privilegios de accesos a usuarios.
REVOKE: suprime privilegios de accesos a usuarios.

3.4-Control de transacciones

COMMIT: finaliza una transacción.
ROLLBACK: aborta la transacción.

3.5-Programático

DECLARE: define un cursor (estructura de control utilizada para el recorrido y potencial procesamiento de los registros del resultado de una consulta).
OPEN: abre un cursor para recuperar resultados de una consulta.
FETCH: es para ir recorriendo los resultados cargados en un cursor.
CLOSE: cierra un cursor.

3.6-Select

El tremendo potencial de SQL se asienta sobre su instrucción más poderosa que es el SELECT. Aquí se dará un pequeño ejemplos de SQL que permitirá al lector mejorar su comprensión sobre las bases de datos relacionales y la sencillez del lenguaje SQL.

Todas las base de datos relacionales tienen un intérprete de instrucciones SQL que se ejecuta sobre el *shell* del SO. Sobre este intérprete, se pueden ejecutar las instrucciones SQL que operarán sobre la base relacional instalada.

En el siguiente ejercicio (Ver Figura 16), se hará una consulta uniendo tres tablas cuyo resultado deberá implicará la instalación de un filtro y un ordenamiento determinado.

Sobre las tablas expuestas en la figura 16, se realizará una consulta que deberá brindar lo siguiente:

Una consulta de las facturas realizadas a los clientes 10200 y 10202 que contenga el número de factura, el código del cliente, la razón social, el importe y el código de vendedor que efectuó la venta. Esta consulta deberá estar ordenada por número de factura. Esta instrucción debería ser de la siguiente forma:

```
SELECT TRANSACCIONES-FACTURACION.factura,
       DATOS-CLIENTES.codigo-cliente, DATOS-CLIENTES.razon-social,
       TRANSACCIONES-FACTURACION.importe,
        ZONAS-GEOGRAFICAS.vendedor

FROM ZONAS-GEOGRAFICAS, DATOS-CLIENTES, TRANSACCIONES-
       FACTURACION

WHERE DATOS-CLIENTES.codigo-cliente =
      TRANSACCIONES-FACTURACION.codigo-cliente

          AND   DATOS-CLIENTES.zona = ZONAS-GEOGRAFICAS.zona

          AND   TRANSACCIONES-FACTURACION.codigo-clientes = 10200

          AND   TRANSACCIONES-FACTURACION.codigo-clientes = 10202

ORDER BY TRANSACCIONES-FACTURACION.factura
```

El resultado de la consulta sería el siguiente:

factura	codigo-cliente	razon-social	importe	vendedor
100	10200	Los cocos	560	3
101	10200	Los cocos	1050	3
102	12002	Los Hornos	30	2
105	12002	Los Hornos	60	2

La explicación de la instrucción es la siguiente:

En el **SELECT**, se enumeran los campos que se quieren listar en el orden que se quieren obtener. El nombre del campo se especifica junto con el nombre de la tabla separada por un punto.

En el **FROM**, se colocan las tablas de donde se obtienen los campos enunciados en el **SELECT** (no importa el orden).

En el **WHERE**, se realizan las uniones de las tablas, es donde se establecen las relaciones. La tabla de DATOS-CLIENTES se une con la tabla TRANSACCIONES-FACTURACION por el campo **codigo-cliente**. La tabla TRANSACCIONES-FACTURACION se une con la tabla ZONAS-GEOGRAFICAS mediante el campo **zona**.

Además en el **WHERE** se establecen los filtros en este caso que el **codigo-cliente** sea igual a 10200 y 10202.

El **ORDER BY** establece el orden en que se generará la consulta, en este caso, por TRANSACCIONES-FACTURACION.factura.

4-¿Qué son las vistas en un DBMS?

Dentro de un sistema de base de datos relacional, se pueden definir vistas. Las vistas no son tablas del sistema, pero el usuario las percibe como tales. Las vistas proporcionan seguridad y funcionalidad. Por ejemplo, cuando exista una consulta muy utilizada que requiere la unión de varias tablas relacionadas, es práctico construir una vista (la que une las tablas relacionadas) y luego la consulta se efectúa, directamente, sobre la vista y no sobre las tablas. La forma más simple de entender una vista es verla como una preconsulta de la base de datos. Las consultas en SQL pueden accionar sobre las vistas, lo que constituye una consulta sobre una consulta. Estas vistas también pueden superponerse entre ellas; es decir, que pueden compartir datos.

Una vista sobre las tablas de la figura 16 puede ser la siguiente:

factura	codigo-cliente	razon-social	importe	vendedor
100	10200	Los cocos	560	3
101	10200	Los cocos	1050	3
102	12002	Los Hornos	30	2
105	12002	Los Hornos	60	2

Este es el **SELECT** del punto 3.6, puede ser definido como una vista y el usuario pasa a ver esta consulta como si fuera una tabla. Estas vistas son muy apropiadas para utilizar herramientas de usuarios finales con funcionalidades para consultar DBMS, como por ejemplo Excel.

Por otro lado, las vistas pueden ser muy útiles para manejar la seguridad. A los usuarios, se les asignan permisos sobre la vistas, con lo cual ya se está restringiendo el acceso a la base de datos a las tablas, las columnas y las filas que contempla la vista.

5-Instalación de un DBMS

La instalación de un DBMS difiere mucho entre los distintos productos (Oracle, DB2, MySQL, SQL Server, Informix, etcétera) y también depende del SO sobre el cual se instale.

En algunos casos, las tablas del DBMS se instalan sobre un único archivo del SO, lo cual constituye una estructura de archivo muy compleja; en otros casos, se asigna para cada tabla un archivo del SO. Puede haber instalaciones que combinan ambas variantes. Incluso, los datos de una tabla de un DBMS puede estar en un archivo del SO y los índices de dicha tabla en otro archivo del SO.

FERNANDO J. MARTINI

TECNOLOGIA DE LA INFORMACION

VOLUMEN 3

EL DESARROLLO DE SOFTWARE

TECNOLOGÍA DE LA INFORMACIÓN
VOLUMEN 3

EL DESARROLLO DE SOFTWARE

Versión: 1.04

Martini, Fernando J.
Tecnología de la Información : el desarrollo de software . - 1a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2014.
v. 4, CD-ROM.
ISBN 978-987-33-4694-1
1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

Martini, Fernando J.
Tecnología de la Información. - 1a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2014.
v.4, CD-ROM.
ISBN 978-987-33-4691-0
1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

ÍNDICE

Capítulo I
Aspectos generales

1-El proceso de creación de software

Se define como proceso al conjunto ordenado de pasos a seguir para llegar a la solución de un problema o la obtención de un producto, en este caso en particular, para lograr un producto de software que resuelva un problema específico.

El proceso de creación de software puede llegar a ser muy complejo, dependiendo de su porte, características y criticidad. Por ejemplo: la creación de un sistema operativo es una tarea que requiere proyecto, gestión, numerosos recursos y todo un equipo disciplinado de trabajo. En el otro extremo, si se trata de un sencillo programa (por ejemplo, la resolución de una ecuación de segundo orden), este puede ser realizado por un solo programador (incluso aficionado) fácilmente.

Considerando los proyectos de gran porte, es necesario realizar complejas tareas, tanto técnicas como de gerencia, una fuerte gestión y análisis diversos (entre otras cosas): La complejidad de ello ha llevado a que se desarrolle una rama de la ingeniería para tratar su estudio y realización: conocida como Ingeniería de Software.

2-La ingeniería de software

Ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software. La creación del software es un proceso intrínsecamente creativo y la ingeniería del software trata de sistematizar este proceso con el fin de acotar el riesgo del fracaso en la consecución del objetivo creativo por medio de diversas técnicas que se han demostrado adecuadas sobre la base de la experiencia.

Un objetivo de décadas ha sido el encontrar procesos y metodologías, que sean sistemáticas, predecibles y repetibles, a fin de mejorar la productividad y la calidad en el desarrollo de productos software. Muchas y variadas metodologías para el desarrollo de software han sido puestas en prácticas con mayor o menor éxito.

3-Etapas del proceso de desarrollo de software

Se consideran las siguientes etapas en el desarrollo de software:

- Comunicación.
  - Análisis de requisitos.
  - Especificación.
- Planeamiento.
- Modelado arquitectónico.
- Construcción.
  - Programación.
  - Prueba.
- Despliegue.

3.1-Comunicación

Es un fuerte intercambio de información entre el cliente y el equipo de desarrolladores.

3.1.1-Requisitos

Extraer los requisitos de un producto de software es la primera etapa para crearlo. Un requisito es una necesidad documentada sobre el contenido, la forma o la funcionalidad de un producto o servicio. Establece qué debe hacer el sistema, pero no cómo hacerlo. Los requisitos deben verificarse en cuanto a su consistencia e integridad. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios.

La captura, el análisis y la especificación de requisitos es una parte crucial; de esta etapa, depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos procesos de trabajo para estos fines. Tan importante es esta etapa que ya se ha comenzado a crear una subrama de la ingeniería de software que se denomina ingeniería de requisitos.

El Instituto de Ingenieros Electricistas y Electrónicos (IEEE por su denominación en inglés: *Institute of Electrical and Electronics Engineers*) creó, y ha sido muy difundida su utilización, una seria de especificaciones para efectuar el relevamiento de requisitos.

3.1.2-Especificación

La especificación de requisitos describe el comportamiento esperado del software una vez desarrollado. La problemática fundamental de esta etapa se centra en cómo chequear con los usuarios, si el comportamiento que los desarrolladores piensan que debe cumplir el software es el mismo que están pensando los usuarios. Para este fin, se han desarrollado modelos de especificación con buenos resultados.

3.2-Planeamiento

La finalidad del planeamiento es crear un guía de trabajo que permita que un gestor de proyectos pueda usarla para acompañar el progreso de su trabajo. Esta etapa sirve para cuantificar el tiempo y los recursos que un proyecto necesitará, además de generar las herramientas que permitirán efectuar un adecuado seguimiento y control.

Inicialmente, el alcance del proyecto es definido y los métodos apropiados para completar el proyecto son determinados (todas las tareas necesarias). La duración para las distintas tareas necesarias para completar el trabajo son listadas y agrupadas en una estructura de descomposición del trabajo. Luego, las dependencias lógicas entre tareas son definidas. Una vez establecido y aceptado el plan, este se convierte en el lineamiento básico. El progreso será medido contra este lineamiento durante toda la vida del proyecto. El planeamiento del proyecto no es algo para hacerse solamente una vez al comienzo del proyecto. Observar el progreso de su equipo y actualizar adecuadamente el plan de proyecto debe ser una tarea constante. Un programa computacional de gestión de proyectos puede ser útil si es usado correctamente.

3.3-Modelado arquitectónico

La integración de la infraestructura de hardware con el desarrollo de aplicaciones, con las bases de datos y las herramientas de gerenciamiento de proyectos requieren de capacidades especiales para poder ser conceptualizados y proyectados a futuro. El rol en el cual se delegan todas estas actividades es el del arquitecto.

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema. Se le ha denominado Arquitectura de Software porque se asemeja mucho a las tareas que realiza un arquitecto al planear las tareas para realizar una construcción. La arquitectura de software define, de manera abstracta, y utilizando modelos específicos de esta rama de la ingeniería, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos.

Los modelos arquitectónicos describen visiones que pueden ser:

- La visión estática: describe qué componentes tiene la arquitectura.
- La visión funcional: describe qué hace cada componente.
- La visión dinámica: describe cómo se comportan los componentes a lo largo del tiempo y cómo interactúan entre sí.

3.4-Construcción

Es la etapa de reducir un diseño a un determinado código de programación. Muchos de los aspectos de la calidad del software tienen que ver con esta etapa, pero no es necesariamente la que demanda mayor trabajo ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

La programación realizada requiere de una exhaustiva prueba. Entre las diversas pruebas que se le efectúan al software, se pueden distinguir principalmente:

- Prueba unitarias: consisten en probar o testear piezas de software pequeñas; a nivel de secciones, procedimientos, funciones y módulos; aquellas que tengan funcionalidades específicas. Dichas pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código, mucho más reducidas que el conjunto, y que tienen funciones concretas con cierto grado de independencia.

- Pruebas de integración: se realizan una vez que las pruebas unitarias fueron concluidas exitosamente; con estas se intenta asegurar que el sistema completo, incluso los subsistemas que componen las piezas individuales grandes del software, funcionen correctamente al operar en conjunto.

Generalmente, existe una fase probatoria final y completa del software, llamada Beta Test, durante la cual el sistema instalado en condiciones normales de operación y trabajo es probado exhaustivamente a fin de encontrar errores, inestabilidades, respuestas erróneas, etcétera, que hayan pasado los controles previos. Los posibles errores encontrados se transmiten a los desarrolladores para su depuración. En general son los usuarios finales (cliente) quienes realizan el Beta Test, teniendo para ello un periodo de prueba pactado con el desarrollador.

3.5-Despliegue

El despliegue del software es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. Constituye la etapa final en el desarrollo propiamente dicho del software. Luego de esta etapa, el producto entrará en la fase de funcionamiento y producción, para el que fue diseñado.

4-Metodología de desarrollo de software

Se llama método (del griego *meta*: «más allá» y *hodos*: «camino, literalmente camino o vía para llegar más lejos») al modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado. Para lograr un producto de software de buena calidad, dentro del presupuesto y a tiempo, es conveniente manejarse con un método preestablecido y probado en su eficiencia.

Una metodología de desarrollo de software es un marco o *framework* de trabajo usado para estructurar, planificar, dirigir y controlar el proceso de desarrollo de un sistema de información. A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad.

El *framework* para metodología de desarrollo de software consiste en:

- Una filosofía de desarrollo de programas de computación.
- Herramientas, modelos y métodos para asistir al proceso de desarrollo de software.

La dificultad propia del desarrollo de software y su impacto en el negocio han puesto de manifiesto las ventajas (y en muchos casos la necesidad) de aplicar una metodología formal para llevar a cabo los proyectos de este tipo. El objetivo es convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de alta calidad, que satisfaga las necesidades y expectativas del cliente. Atrás debe dejarse el modo de trabajo artesanal, que a menudo requiere de esfuerzos heroicos para llegar a buen puerto, con los consecuentes desfases de fechas y costos, y el más que probable desgaste personal del equipo de proyecto.

La metodología que se adopte dependerá del las características del proyecto (urgencia, tamaño, producto, etcétera), las habilidades del equipo de trabajo, el lenguaje de programación a adoptar, etcétera.

5-Enfoques de las diversas metodologías de ingeniería de software

Existen tres enfoques básicos para el desarrollo de software. Sobre alguno de estos tres enfoques, se asientan las distintas metodologías de desarrollo. Algunas metodologías combinan estos enfoques básicos.

Los tres enfoques son: lineal o en cascada, incremental e iterativo.

5.1-El proceso lineal o en cascada

El proceso lineal o en cascada es el menos aconsejable y se pude decir que es prácticamente imposible de utilizar, salvo en proyectos muy pequeños. El proceso lineal consiste en tomar el sistema a desarrollar en forma integral, realizando cada una de las cinco etapas del desarrollo del software y considerando la totalidad del sistema y hasta el último de sus detalles (Ver Figura 1). Es decir, realizar la comunicación (requisitos y especificaciones) total del sistema a desarrollar; luego, pasar a la de planeamiento de todo el proyecto; después, a la de arquitectura de todo el sistema; luego, a la de construcción del total; y por último, a la de despliegue completo. Para comenzar una nueva etapa, deber estar totalmente terminada la anterior.

Las características de este modelo son:

- El producto resultante solo se ve al final de todo el proyecto, por lo tanto, cualquier error en los procesos se detecta muy tardíamente.
- No atiende la naturaleza cambiante de las organizaciones, ya que la demora en entregar el producto puede dejarlo muy desactualizado de la nueva situación.

- Es muy difícil que los usuarios puedan establecer los requisitos de todo el sistema en una única etapa.

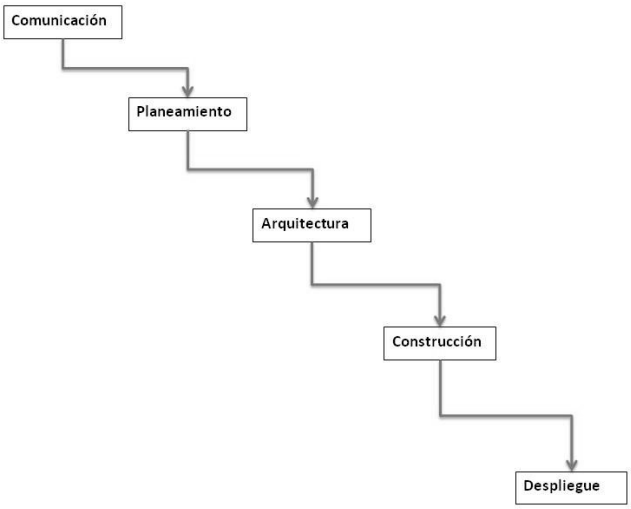


Figura 1

5.2-El proceso incremental

El modelo incremental se caracteriza por construir el sistema por módulos. La modularidad es una necesidad para que un sistema pueda ser construido e implementado por etapas. Consiste en dividir un sistema en subsistemas. La modularidad requiere de una división de los sistemas en partes que después sea simple ensamblarlas. Cada incremento agrega un nuevo módulo.

El sistema se desarrolla como una secuencia de módulos que se entregan en su versión definitiva.

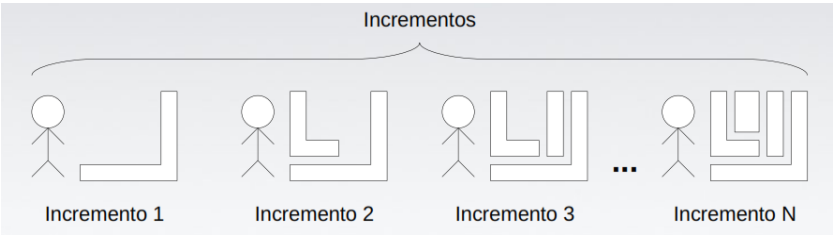


Figura 2 (Universidad de los Andes)

Las características de este modelo son:

- Cada incremento agrega módulos.
- La propuesta del modelo es diseñar sistemas que puedan entregarse por piezas (módulos).
- Los usuarios no tienen que esperar largos periodos de tiempo para empezar a recibir parte del sistema. Esto aviva la participación del usuario en todo el proyecto.
- No es necesario terminar un módulo para comenzar el otro. Por ejemplo: un módulo puede estar en la etapa de comunicación, otro en el de diseño arquitectónico y otro en el de despliegue. El desarrollo de módulos se realiza solapando unos con otros.

5.3-El proceso iterativo

La idea principal detrás de mejoramiento iterativo es desarrollar un sistema de programas de manera incremental, permitiéndole al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema. El aprendizaje viene de dos vertientes: el desarrollo del sistema, y su uso (mientras sea posible). Los pasos claves en el proceso son comenzar con una implementación simple de los requisitos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios mejorando el diseño anterior y se agregan nuevas funcionalidades y capacidades al sistema (Ver Figura 3).

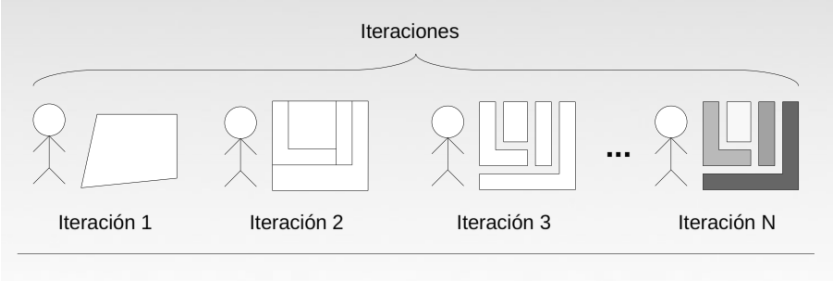


Figura: 3 (Universidad de los Andes)

Las características de este modelo son:

- Logran su objetivo por medio del desarrollo de un producto que evolucionan a medida que se tiene realimentación del cliente.
- Requieren de un cliente involucrado durante todo el curso del proyecto.
- Infunden responsabilidad en el equipo de desarrollo al trabajar directamente con el cliente.
- Técnicamente se sustentan en el uso de prototipos.

6-El prototipo

Los prototipos son una versión limitada de un producto, que permite a las partes (desarrollador y cliente) probarlo en situaciones reales o explorar su uso, creando así un proceso de diseño de iteración que genera calidad.

Son muy útiles cuando los usuarios no tienen una idea definitiva del producto. Su utilización es muy provechosa cuando el desarrollo de software combina desarrollos incrementales con iterativos.

El problema de los prototipos es que muchas veces el prototipo no difiere mucho de la versión final. Esto ocurre porque la funcionalidad no es posible de simplificar en una versión limitada. En estos casos, conviene atacar la versión definitiva sin intentar un prototipo previo.

Capítulo II

Modelado de requisitos y especificaciones (análisis)

1-¿Qué es la ingeniería de requisitos?

La ingeniería de requisitos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o a modificar, tomando en cuenta los diversos requisitos de los interesados y teniendo en cuenta que pueden entrar en conflicto entre ellos.

El propósito de la ingeniería de requisitos es hacer que los mismos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables, no deben presentar ambigüedades o contradicciones.

El producto de la ingeniería de requisitos es la redacción de una forma entendible para los usuarios de los requisitos que deberá alcanzar el sistema. Para este objetivo, existen diversos modelos, como: escenarios de uso, lista de funciones y características, especificaciones, etcétera.

La tarea más difícil en el desarrollo de software es determinar con la mayor exactitud posible lo que deberá hacer el sistema. Una mala especificación de un requisito es muy difícil de solucionar cuando el producto está terminado.

Desde un punto de vista conceptual, las actividades de la ingeniería de requerimientos son seis:

- Obtener requisitos: a través de entrevistas o comunicación con clientes o usuarios, para saber cuáles son sus expectativas.
- Elaboración de requisitos: detectar y corregir las falencias comunicativas, transformando los requisitos obtenidos de entrevistas y requisitos en condiciones apropiadas para ser tratados en el diseño. Esto requiere de una tarea de modelado que refina la calidad de los requisitos.
- Solución de conflictos: es común que clientes o usuarios entren en conflicto con la solución deseada. El analista debe resolverlos valiéndose de diversas técnicas.
- Especificar: para documentar los requisitos, se pueden utilizar plantillas estandarizadas, redactar documentos en lenguaje natural o utilizar modelos gráficos, incluso pueden usarse prototipos.
- Validar los requisitos: comprobar que los requisitos implementados se corresponden con lo que inicialmente se pretendía.
- Gestión de requisitos: el cambio es una constante en las organizaciones y esos cambios deben ser reflejados en sus sistemas. Poder efectuar una efectiva trazabilidad de los requisitos es esencial para que el sistema esté durante toda su vida efectivamente controlado. Debe quedar registrado: dónde se originaron los requisitos, quiénes los generaron, los cambios a los que fueron sometidos, por qué se cambiaron, quién solicitó los cambios, etcétera.

2-Casos de uso

Un caso de uso es una técnica para documentar los requisitos del sistema Un caso de uso se materializa con todas las formas de emplear un sistema para alcanzar una meta particular para un usuario particular.

Un caso de uso es:

- Una secuencia de acciones que un sistema ejecuta y que arroja un resultado observable de valor para un usuario particular.
- El comportamiento específico de un sistema, que participa en una colaboración con un usuario para entregar algo de valor para ese usuario.
- La unidad más pequeña de actividad que proporciona un resultado significativo para el usuario.
- El contexto para un conjunto de requisitos relacionados.

Un caso de uso está integrado por un escenario y sus actores. Un escenario especifica las acciones e interacciones que el usuario realiza dentro de él. Un actor es algo que se comporta dentro del escenario (persona u otro sistema) y que está identificado por un rol.

Para entender un caso de uso narramos historias. Las historias cubren tanto la forma como se alcanza la meta de manera exitosa y la forma como se maneja cualquier problema que ocurre en el camino. Las historias nos ayudan a entender el caso de uso y lo implementan porción por porción.

<u>Ejemplo de la narración de un caso de uso</u>

Caso de uso:	pago en comercios minoristas con teléfono celular.
Actor primario:	comprador.
Actor secundario:	vendedor.

Objetivo del sistema:	pagar bienes o servicios mediante la utilización de un teléfono celular.
Condiciones previas:	el cliente debe haber ingresado fondos en su cuenta de su proveedor de pagos telefónicos.
Activador:	el cliente activa el sistema comunicándose al servicio telefónico de pagos ingresando 4 dígitos.
Escenario 1 - el cliente:	ingresa los 4 dígitos del servicio de pagos telefónicos.
Escenario 2 – el cliente:	recibe una pantalla que solicita el código del vendedor y el importe a pagar.
Escenario 3 - el cliente:	ingresa el código de PIN.
Escenario 4 - el cliente:	ingresa el código del vendedor.
Escenario 5 - el cliente:	ingresa el importe a pagar.
Escenario 6 - el cliente:	recibe por SMS una confirmación del pago.
Escenario 7 - el vendedor:	recibe por SMS una confirmación del pago.
Excepción 1:	no se puede conectar al servicio de pago telefónico, el cliente debe volver a intentar.
Excepción 2:	número de vendedor incorrecto, debe volver a ingresarlo.
Excepción 3:	ingresa el PIN incorrectamente, el cliente debe volver a intentarlo. Se admiten dos intentos fracasados. Al tercer intento, se desactiva el PIN.
Excepción 3:	el pago no es aceptado por saldo insuficiente.
Prioridad:	esencial.
Disponible:	desde el primer incremento.
Frecuencia de uso:	muchas veces al día.

3-UML

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un «plano» del sistema (modelo), incluyendo aspectos conceptuales, tales como procesos de negocio, funciones del sistema y aspectos concretos, como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un «lenguaje de modelado» para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descripto el modelo.

UML no especifica una determinada metodología para usar.

4-Diagrama de casos de uso

En el Lenguaje de Modelado Unificado, un diagrama de casos de uso es una especie de diagrama de comportamiento. UML define una notación gráfica para representar casos de uso, llamada modelo de casos de uso. Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso.

Un diagrama de casos de uso es una forma simple de presentar una visión general de los requisitos de un sistema. Se pueden ver todas las formas en que el sistema se puede usar, quién inicia la interacción y todas las partes involucradas.

Ejemplo de diagrama de casos de uso

El diagrama de casos de uso de la figura 4 describe la funcionalidad de un sistema telefónico muy simple. Los casos de uso están representados por elipses y los actores por figuras humanas.

La interacción entre actores no se ve en el diagrama de casos de uso. UML define una notación gráfica para realizar diagramas de casos de uso, pero no el formato para describir el caso de uso.

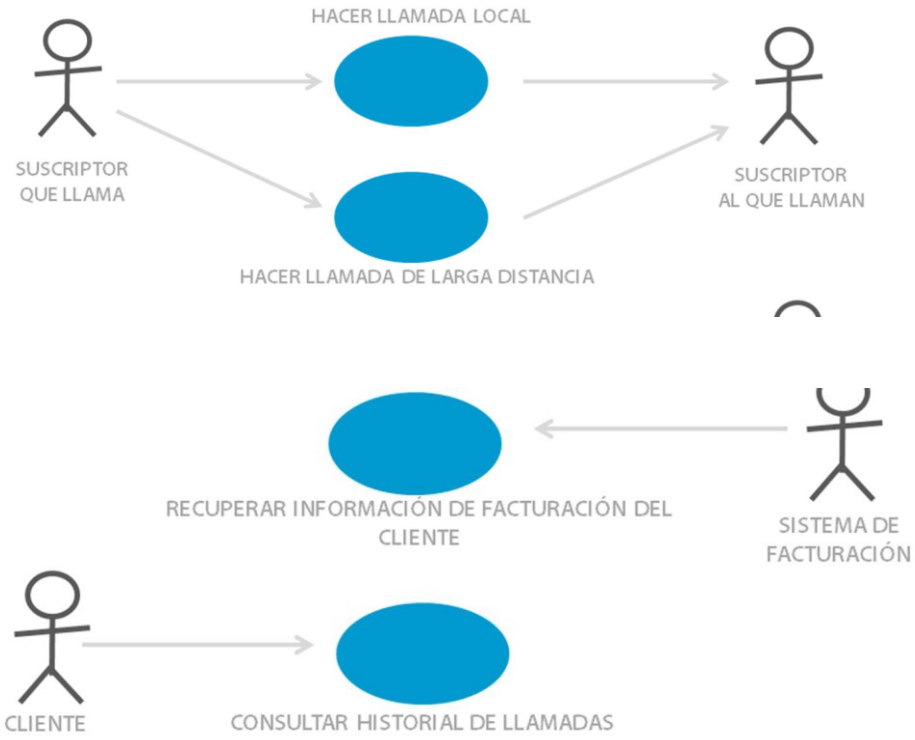


Figura 4

4.1-Diagrama de actividad (complementa el modelo de caso de uso)

El diagrama de actividades complementa el modelo de caso de uso mostrando las interacciones dentro del escenario. Este diagrama utiliza rectángulos redondeados para graficar las funciones del sistema, flechas para el flujo de actividades y rombos para bifurcar por una decisión.

El diagrama de actividad para el caso de uso del sistema de pagos por celular se encuentra graficado en la figura 5.

4.2-Diagramas de carril (complementa el modelo de caso de uso)

Este diagrama de UML es una variante del diagrama de actividad que permite ver con un poco más de claridad la actividad de cada uno de los actores del caso de uso. En la figura 6, se desarrolla el diagrama de carril del caso de uso desarrollado.

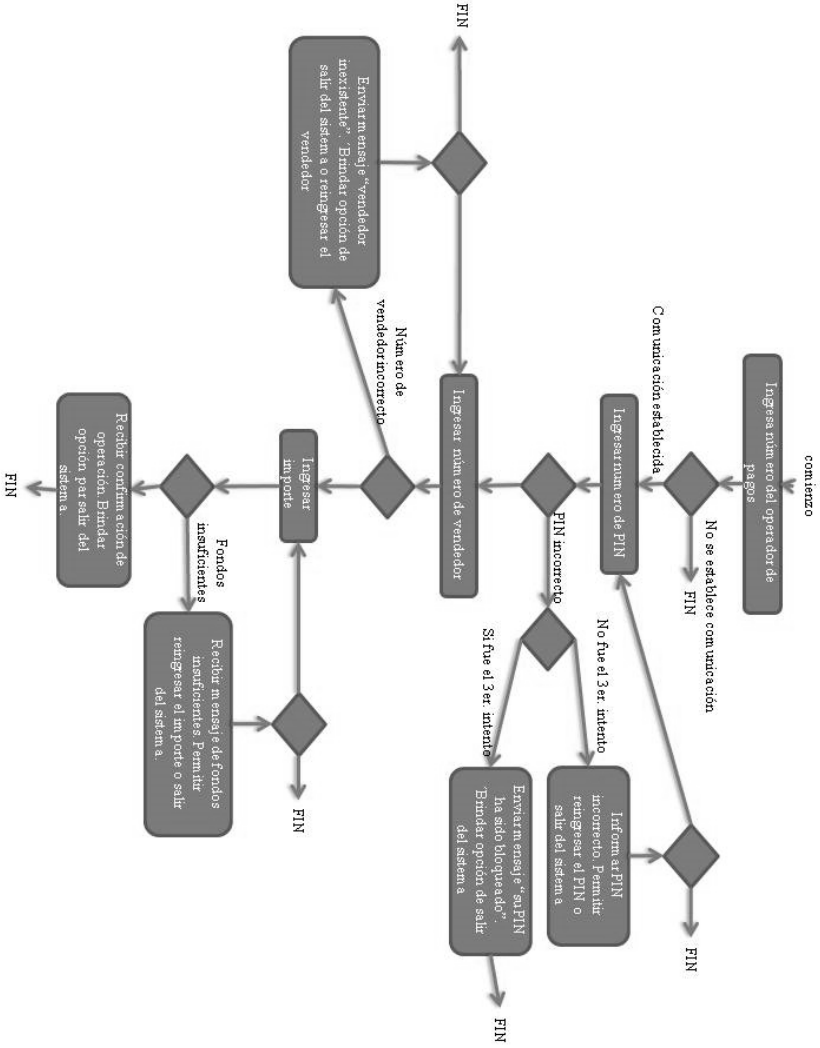

comienzo
Ingresa número del operador de pagos
Comunicación establecida
No se establece comunicación
FIN
Ingresar número de PIN
PIN incorrecto
No fue el 3er. intento
Informar PIN incorrecto. Permitir reingresar el PIN o salir del sistema
Si fue el 3er. intento
Enviar mensaje "su PIN ha sido bloqueado". Brindar opción de salir del sistema
Ingresar número de vendedor
Número de vendedor incorrecto
Enviar mensaje "vendedor inexistente". Brindar opción de salir del sistema o reingresar el vendedor
Ingresar importe
Fondos insuficientes
Recibir mensaje de fondos insuficientes. Permitir reingresar el importe o salir del sistema.
Recibir confirmación de operación. Brindar opción para salir del sistema

Figura 5

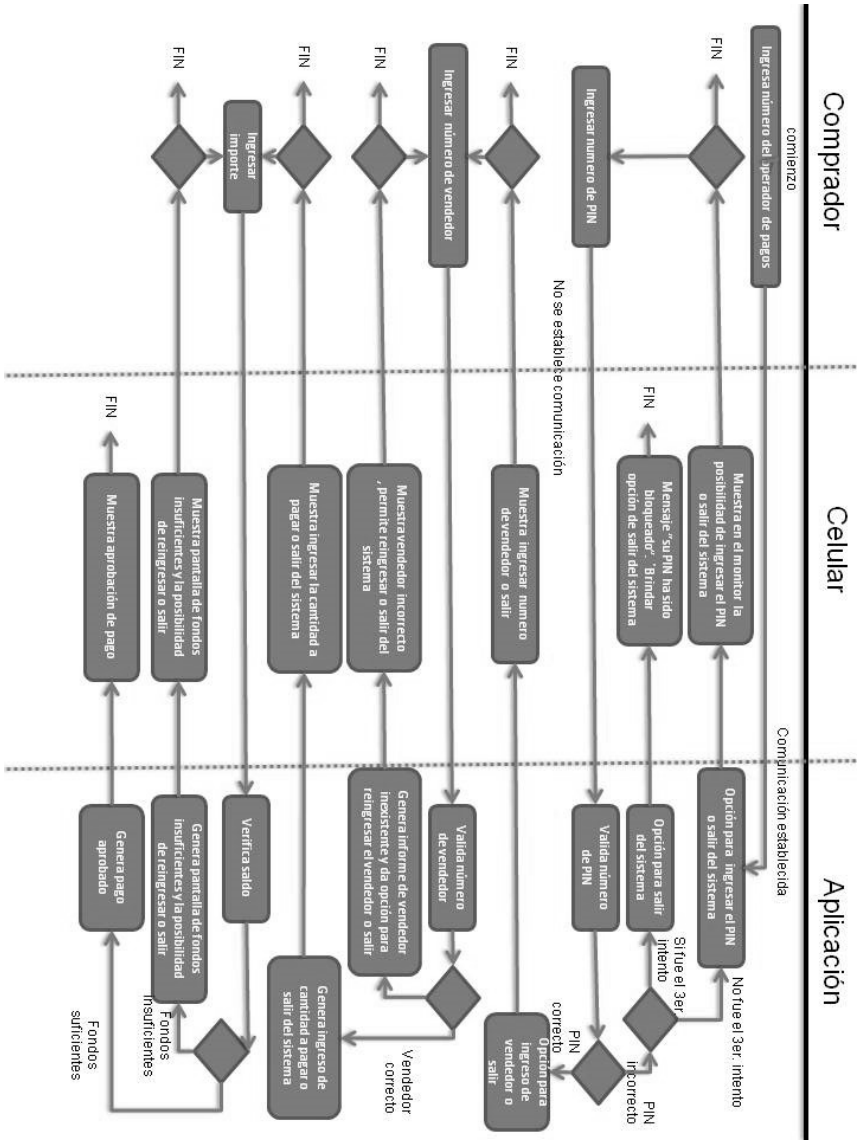


Figura 6

5-Modelo basado en el flujo de datos

El diagrama de flujo de datos o DFD permite realizar un modelado de la información junto a sus funciones. El refinamiento del DFD implica un acercamiento hacia los procesos que en definitiva contendrán las aplicaciones. El DFD se construye en varios niveles, cada nivel va descomponiendo burbujas del nivel superior. Siempre un nivel inferior tiene una burbuja en el nivel superior.

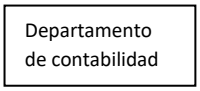
El DFD se construye con los siguientes componentes:
Burbuja: es un proceso al que también se lo puede llamar función o transformación. La burbuja transforma entradas en salidas.

Flujo: se representa con una flecha que entra o sale de una burbuja. Se usa para graficar el movimiento de paquetes de información. Son datos en movimiento.

Almacén: es una colección de datos en reposo.

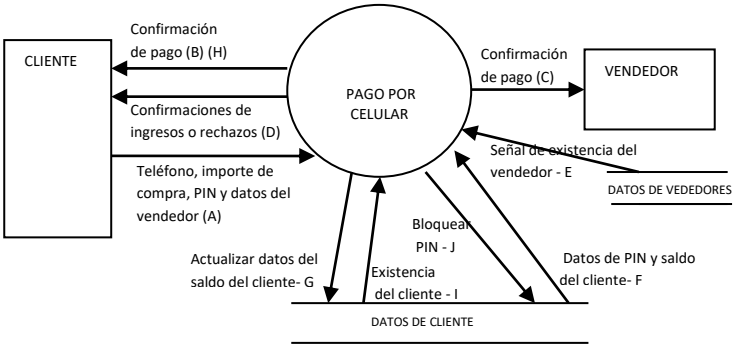
PEDIDOS

Terminadores: se representa con un rectángulo y son entidades externas al sistema con el cual el sistema se comunica. Producen información para el sistema o son receptoras de información del sistema. En la mayoría de los libros clásicos sobre análisis estructurado los terminadores no deben utilizarse en los DFD (porque consideran a éstos fuera del dominio del cambio, es decir que los terminadores no pueden ser modelados, son externos y el analista con ellos no puede hacer nada). En este libro se utilizan considerando que los mismos aportar claridad a la comprensión del modelo.



El primer nivel se denomina diagrama de contexto y debe estar formado por una sola burbuja que identifica el sistema completo con sus correspondientes flujos de datos que ingresan al sistema y los que salen del sistema, desde y hacia entidades externas (terminadores).

Por ejemplo, el diagrama de contexto del caso de uso descripto sería:



Luego del diagrama de contexto, se realiza lo que se denomina DFD de nivel 0:

Una buena forma de construir el nivel 0 del DFD es analizar la narrativa de lo que ha requerido el usuario en la primer entrevista.

El cliente ingresa al sistema mediante la utilización de su celular llamando a un número de tres cifras más numeral. El cliente puede ser rechazado. Si el cliente es rechazado, deberá ser informado y se le brindará la opción de salida del sistema. El servicio de pago deberá ofrecer una función para que el cliente ingrese su PIN, este PIN deberá ser validado por el sistema. Si el PIN ingresado es correcto, deberá pasar a permitir ingresar el número de vendedor; de lo contrario, deberá mostrar una pantalla al cliente que le permita ingresar nuevamente el PIN o salir del sistema. Esta posibilidad debe ser otorgada en tres oportunidades. Si en la tercera oportunidad también es rechazado, el sistema deberá informar al cliente que su cuenta ha sido bloqueada y dará la opción de salir del sistema. Si el PIN es aceptado, se pasa a la función de permitir el ingreso del código del vendedor; si el código resulta ser incorrecto, se le permitirá al cliente ingresarlo tantas veces como quiera o brindarle la opción de salir del sistema. Si el vendedor es correcto, el sistema pasará a permitir ingresar el importe a comprar. El sistema validará el importe y si el saldo del cliente fuese insuficiente, se le informará al cliente permitiéndole rectificar el importe tantas veces como lo desee o le dará la opción de salir del sistema. Si el importe es aprobado, se le deberá restar al cliente el importe de su saldo y enviarle un mensaje de que la transacción fue aprobada y le brindará la opción de salir del sistema. Luego el sistema de servicio de pagos telefónicos deberá enviar un SMS tanto al cliente como al vendedor confirmando la aprobación de la compra.

De esta narrativa, se deben aislar todos los sustantivos y verbos. Los verbos son los procesos, los sustantivos son entidades externas.

En la figura 7, se muestra el primer intento de un DFD de nivel 0. Cada DFD no debería tener más de 5 o 6 burbujas para ser entendible. Si se necesita un mayor grado de definición, se debería realizar un DFD de nivel inferior y así sucesivamente. Los niveles inferiores son como explosiones de cada una de las burbujas de nivel superior. Como el lector podrá ver, el DFD inicial (Ver Figura 7) ha quedado sumamente complicado, por lo que conviene darle un mayor nivel de simplificación agrupando burbujas. La tarea de establecer correctamente el nivel de DFD 0 es crucial para que el desarrollo completo sea claro y comprensible. En la figura 8, se muestra el definitivo nivel 0. Recién en ese momento, se puede empezar a descomponer cada una de las burbujas en niveles inferiores. En la figura 9, se desarrolla el nivel 1 de la burbuja 1. En la figura 10, se desarrolla el nivel 1 de la burbuja 2. En la figura 11, se desarrolla el nivel 1 de la burbuja 3. En la figura 12, se desarrolla el nivel 1 de la burbuja 4. De las burbujas 5 y 6, no se requiere mayor nivel de detalle.

Es muy importante que al pasar de nivel se verifique la consistencia de entradas y salidas de la burbuja superior con las entradas y salidas del nivel inferior.

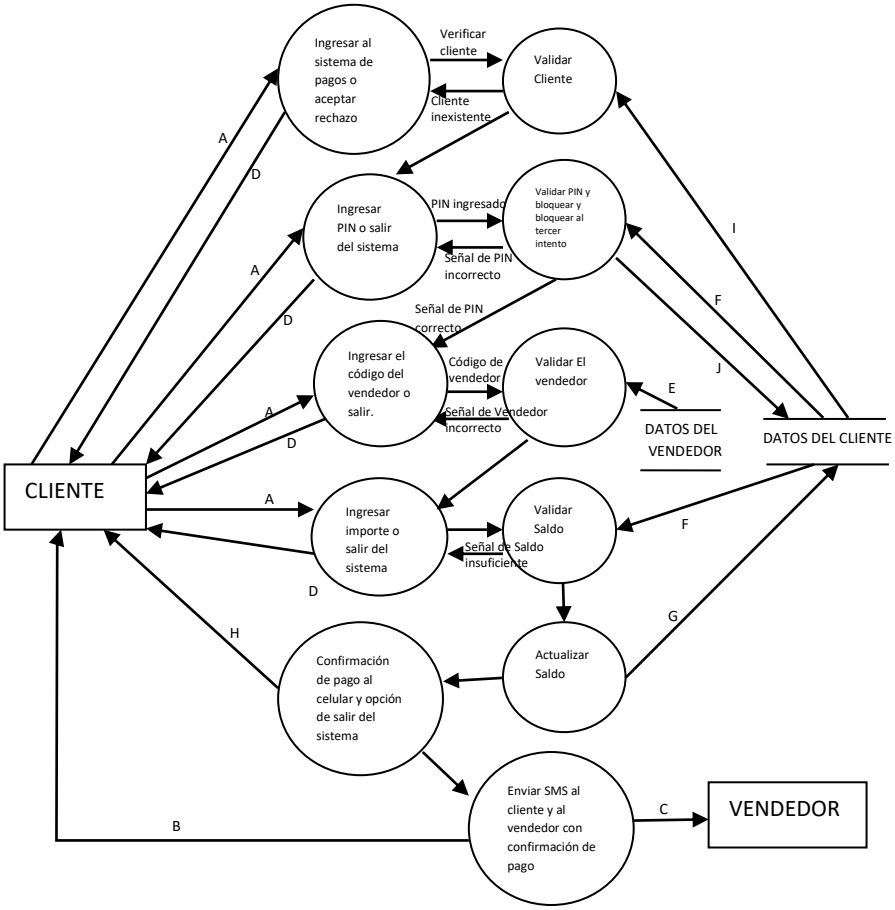

Ingresar al sistema de pagos o aceptar rechazo
Verificar cliente
Validar Cliente
Cliente inexistente
Ingresar PIN o salir del sistema
PIN ingresado
Validar PIN y bloquear y bloquear al tercer intento
Señal de PIN incorrecto
Señal de PIN correcto
Ingresar el código del vendedor o salir.
Código de vendedor
Validar El vendedor
Señal de Vendedor incorrecto
DATOS DEL VENDEDOR
DATOS DEL CLIENTE
CLIENTE
Ingresar importe o salir del sistema
Validar Saldo
Señal de Saldo insuficiente
Confirmación de pago al celular y opción de salir del sistema
Actualizar Saldo
Enviar SMS al cliente y al vendedor con confirmación de pago
VENDEDOR
A
D
A
D
A
D
A
D
E
F
I
J
F
G
H
C
B

Figura: 7

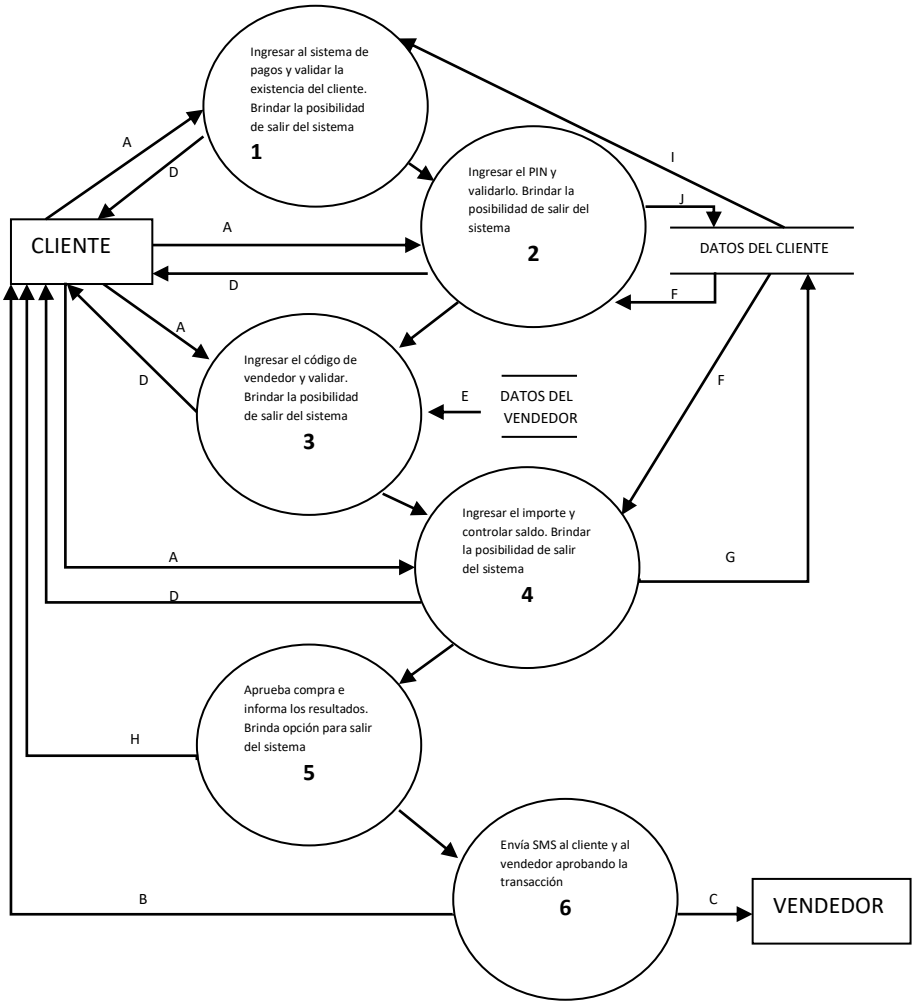


Figura 8

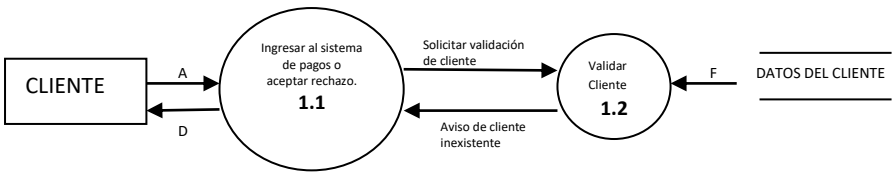


Figura 9

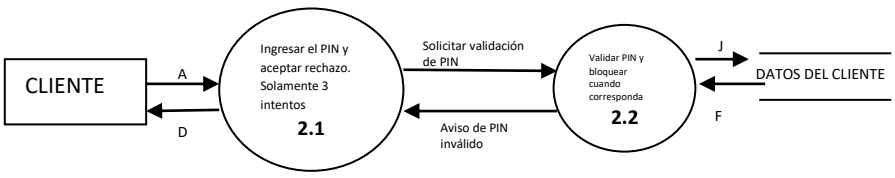


Figura 10

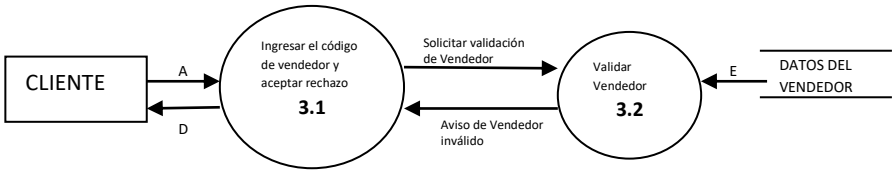


Figura 11

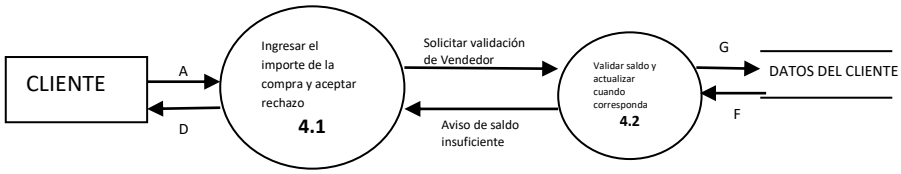


Figura 12

5.1-Especificación de proceso

Es una herramienta de modelado de sistemas, que permite definir qué sucede en los procesos o funciones de un sistema (qué sucede en cada burbuja de nivel inferior). El objetivo es definir qué debe hacerse para transformar ciertas entradas en ciertas salidas. Para estas especificaciones, existen muchas técnicas; pero las más utilizadas son: la tabla de decisiones y el lenguaje de diseño de procesos (LDP). El lenguaje de diseño de procesos es prácticamente un especificación de lo que debe hacer el programa, que es prácticamente escribir el programa en lenguaje de humano y no en un lenguaje de alto nivel.

En el caso de la burbuja 2.1, un ejemplo de LDP sería:

Para que el cliente valide el PIN el celular deberá mostrar en el visor la posibilidad de ingresar 6 caracteres numéricos además de mostrar un mensaje que diga "ingrese el PIN XXXXXX". Además debe brindarse la opción de "SEND para enviar" y "# para finalizar". Al recibir la información de la burbuja 2.2 mostrar: si el PIN fue aprobado "ingreso aceptado" y pasar a función 3.1. Si no fue aceptado y no fue el tercer intento mostrar "PIN incorrecto ingréselo nuevamente XXXXXX" con las mismas opciones de SEND y # anteriores. Si fue la tercera oportunidad mostrar un mensaje que diga "su PIN ha sido bloqueado comuníquese con LA EMPRESA" y brindar la opción de salir del sistema con #.

6-Modelado basado en clases

Suele decirse que la diferencia entre el modelado basado en flujo de datos y el modelado basado en el clases es que el primero se enfoca en las funciones que deberá cumplir el sistema, mientras que el segundo se estructura poniendo el acento en los objetos (factura, nota de pedido, celular, punto de venta, etcétera). Esta definición no aclara mucho y es confusa, ya que en definitiva los objetos realizan funciones (y funciones específicas de cada objeto); por lo tanto, la estructuración del sistema basado en clases también es funcional.

Para entender el modelo basado en clases, primero hay que saber la diferencia que existe entre una forma de programar estructurada que surge de los modelos basados en flujo de datos y la forma de programar orientada a objeto que surge de la modelación basada en clases. El modelado basado en clases es una necesidad para poder programar con orientación a objetos.

6.1-Programación tradicional o estructurada basada en el modelo de flujo de datos.

Un programa estructurado consiste en un conjunto de instrucciones, cuyo objetivo está directamente ligado a resolver una funcionalidad general de la aplicación, como por ejemplo, un programa de facturación. Si la aplicación requiere un programa para facturar, se especifica el proceso y un programador toma esa especificación y la resuelve con algún lenguaje específico de programación. Generalmente, se desarrolla solamente en un programa que solucione la necesidad de facturar para el problema en concreto.

Para entender mejor cómo se haría un programa estructurado, se verá a continuación el español estructurado o seudocódigo.

6.1.1-Español estructurado

La estructura narrativa del español estructurado es de la siguiente forma:

SI *condición1*
ENTONCES *acción1*
SI NO (no *condición1*)
ENTONCES *acción2*
.

El punto por debajo del SI es muy importante, ya que identifica la conclusión del SI.
Por ejemplo, en un SI anidado.

SI *condición1*
ENTONCES acción1
SI *condición2*
ENTONCES acción2
.
SI NO (*no condición1 o no condición2*)
ENTONCES *acción3*
.

En este caso, se está programando que si se da la *condición1*, se realiza la *acción 1*, y si además de darse la *condición1* se da la *condicion2*, se realiza la *acción2.* El punto es el inicio de la continuación del programa cuando se de la negación de la *condición1* o la negación de la *condición2*. De no poner el punto, el SI NO corresponde solo a la salida de la negación de la *condición2,* desde luego que cuando la *condición1* sea verdadera.

a) En el español estructurado, los verbos siempre se escriben en infinitivo.
b) Las instrucciones se ejecutan una tras otra en forma descendiente, con la excepción que a continuación se verá.
c) Las instrucciones se separan por RÓTULOS que deben ser lo más sintéticos posible y claros sobre la tarea que se está desarrollando dentro del rótulo.

Por ejemplo:

CALCULAR-DESCUENTO
RESTAR descuento a factura-total
IMPRIMIR descuento
CALCULAR-IVA
MULTIPLICAR 0,21 a factura-total para obtener IVA
SUMAR IVA a factura-total para obtener el total de la factura
FIN

d) La excepción para seguir las instrucciones línea tras línea es la instrucción HACER y su variante HACER HASTA condición1.

Por ejemplo, el juego de instrucciones anteriores podría haberse realizado igualmente de la siguiente forma:

HACER CALCULAR-DESCUENTO
HACER CALCULAR-IVA
FIN
CALCULAR-DESCUENTO
RESTAR descuento a factura-total
IMPRIMIR descuento
FIN-CALCULAR-DESCUENTO
CALCULAR-IVA
MULTIPLICAR 0,21 a factura-total para obtener IVA
SUMAR IVA a factura-total para obtener el total de la factura
FIN-CALCULAR-IVA

La instrucción HACER ejecuta el rótulo especificado, es decir que se ejecutan las instrucciones de CALCULAR-DESCUENTO hasta llegar al siguiente rótulo, en este caso: FIN-CALCULAR-DESCUENTO. Luego, retorna a la instrucción de HACER y continúa con la siguiente instrucción, en este caso, otro HACER CALCULAR-IVA que se ejecutará hasta el rótulo FIN-CALCULAR-IVA.

El HACER rótulo-1 HASTA condición1 es una instrucción que ejecuta el rótulo1 tantas veces como sea necesario hasta que se dé la condición1.

Por ejemplo, para imprimir la tabla de multiplicar del 2 al 10, se podría especificar de la siguiente forma:

TABLA_DEL_DOS
MOVER 2 a la variable multiplicando
HACER TABLA-DEL-DOS HASTA multiplicador = 10 iniciando la variable multiplicador en 0 hasta que llegue a 10. → Esta instrucción significa que en cada iteración del rótulo TABLA-DEL-DOS la variable multiplicador se incrementará en 1.
FIN
TABLA-DEL-DOS
MULTIPLICAR multiplicando por multiplicador y dejar el resultado en producto
IMPRIMIR "multiplicando multiplicador = producto"
FIN_TABLA_DEL_DOS

Este programa ejecutará el rótulo TABLA-DEL-DOS 10 veces.

6.1.2-Español estructurado para el caso del ejercicio del DFD "Compra por celular"

Si bien faltan pasos de diseño para llegar del DFD al seudocódigo, es posible improvisar algún ejemplo que dé entendimiento a un lenguaje que sigue el flujo de la información (tradicional).

Esta aplicación está desarrollada en una arquitectura de procesamiento cliente-servidor, por lo que en este caso solamente se programará, en español estructurado, las burbujas del servidor 1.2, 2.2, 3.2 y 4.2.

No es necesario que el lector siga el programa completo, bastará con que advierta el funcionamiento de la lógica del flujo de datos.

En las figuras 13a, 13b, 13c, 13d, 13e y 13f, se podrá seguir el programa línea por línea.

Este programa está armado en tres grandes bloques que son los incluidos en los HACER entre la línea 1 y 6 del rótulo PROCESO.

La instrucción HACER INICIAR-ATENDEDOR no debe realizar ningún bucle, por lo tanto, no hay necesidad de establecer una condición de salida del bloque.

El bloque HACER PROCESO-PRINCIPAL es un bucle con condición de salida que es la activación de la señal de fin-de-proceso = 1. Dentro de este proceso, se realizará la espera de la llamada entrante, las validaciones de quien quiera efectuar un pago y la realización del pago.

EL último bloque es HACER CIERRE y no ha sido desarrollado en este ejemplo.

Siguiendo el flujo del proceso el programa al iniciarse, ingresará en el bloque INICIAR-ATENDEDOR (línea 2). En este bloque, se activará el atendedor con la instrucción LANZAR (línea 9). La línea 11 se utiliza para preguntar por el estatus del atendedor. Si es correcto, es decir que el atendedor está funcionando, el programa sale del bloque INICIAR-ATENDEDOR y continúa. De lo contrario, mandará un mensaje de error a la consola de la computadora y cancelará el programa. En este caso, habrá que ver por qué motivo el atendedor no inició su actividad.

Si el programa sale de INICIAR-ATENDEDOR correctamente, el flujo del programa pasa al bloque de PROCESO_PRINCIPAL que contiene varios bloques (ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE, VALIDAR-PIN, VALIDAR-VENDEDOR, INGRESAR-IMPORTE) en 3 SI anidados. La salida de cada bloque se dará porque: o se cancela el proceso (por decisión de operador de consola, esto ocurre solo en el bloque ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE), o se cancela la llamada (por diversos motivos que ya se verán), o porque el proceso sale ok y pasa al siguiente bloque.

El primer bloque es ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE, cuya condición de salida puede ser por: 1) que el cliente sea válido (por tal motivo, la línea 19 limpia previamente la condición de salida); 2) que se cancele la llamada y por tal motivo se mueva un 1 a fin-de-llamada (por tal motivo, la línea 18 limpia la condición de salida fin-de-llamada); 3) porque se solicite la cancelación del proceso por parte del

operador de consola (esta salida es fin-de-proceso = 1). Como estas tres condiciones en esta primera entrada están en 0, el flujo del programa ingresa en este bloque (ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE línea 20).

Dentro del bloque ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE, las instrucciones de la línea 41 y 42 son para darle oportunidad al operador para que con la tecla ESC pueda salir del programa. Como el programa estará prácticamente siempre haciendo un bucle dentro de ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE, si, en algún momento, el operador oprime la tecla ESC, el programa mandará la señal 1 a fin-de-proceso (línea 42). Entonces, el flujo del programa intentará hacer nuevamente ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE, pero esta vez por la condición de salida activada (fin-de-proceso igual a 1), saldrá para continuar con el flujo del programa. A la salida del bloque ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE, existen unas condiciones (líneas 21 y 22) que analizan si la salida fue por fin-de-llamada o fin-de-proceso (en realidad, pregunta si la salida fue por cliente-valido, que implicaría que no fue ni por fin-de-llamada ni por fin-de-proceso). Si es por fin-de-llamada, se continuará dentro del bucle de PROCESO_PRINCIPAL a la espera de una nueva llamada. Si la salida es por fin-de-proceso, se saldrá del bucle de PROCESO_PRINCIPAL para ir al bloque de CIERRE (línea 5).

Si el flujo del programa continúa por la línea 44, se analiza si la llamada es de un cliente existente (línea 45). Si el cliente no existe, se continúa por línea 49, 50 y 51 que envían aviso de cancelación a quien intentó la llamada, se corta la llamada y se activa la señal de fin-de-llamada para salir del bloque ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE. El programa no continuará con el resto de los bloques de PROCESO_PRINCIPAL (salteará las líneas de la 23 hasta la 37), pero reiniciará el flujo dentro de este proceso, es decir, se quedará a la espera de una nueva llamada.

Si dentro del bloque ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE el cliente existe, el flujo del programa continúa por la línea 45 que envía al celular el mensaje de solicitud de ingresar PIN o brindar la posibilidad de salir del sistema. Luego en la línea 48 activa la señal de cliente-valido.

Si ESPERAR-LLAMADA-Y-VALIDAR-CLIENTE (línea 20) sale del bucle por cliente-valido igual a 1 (línea 23) el flujo del programa continuará por la línea 24. En la línea 25 limpia la condición de salida de validar PIN (además limpia una señal que se utilizará dentro del bloque de VALIDAR-PIN) y luego en la línea 26 ingresa en el bloque de VALIDAR-PIN.

En la línea 56 se esperan 30 segundo para obtener la respuesta del cliente (el PIN o ESC), pasado este tiempo mueve 1 fin-de-llamada, cancela la llamada y sale de este bloque. Por línea 27 (como s-PIN continúa en 0) salteará desde la línea 28 hasta la 37 para iniciar nuevamente el bloque de PROCESO_PRINCIPAL.

Si la respuesta del cliente es un ESC, entonces mueve 1 fin-de-llamada (línea 79), cancela la llamada y sale de este bloque. Por línea 27 salteará desde la línea 28 hasta la 37 (s-PIN continua en 0) para iniciar nuevamente el bloque de PROCESO_PRINCIPAL.

Si la respuesta es un número de PIN incorrecto (línea 68), se analiza si fue el tercer intento (línea 69). Si fue tercer intento, se envía mensaje de bloqueo de cliente, y se cancela la llamada y se mueve 1 a fin-de-llamada (esto ocurre cuando la variable tercer-intento = 2). Por esta última condición sale del bloque VALIDAR-PIN y saltea las líneas 28 a 37 para continuar dentro del bloque PROCESO_PRINCIPAL. Si el PIN es incorrecto, pero no fue el tercer intento, se dejan todas las condiciones de salidas del bloque en 0, se suma uno a tercer-intento (campo que se limpió en la línea 25) y se continúa por la línea 56 a la espera de un nuevo PIN.

Si el PIN es correcto, se activa la señal s-PIN que hace que se salga del bloque VALIDAR-PIN y continuar por línea 28. En la línea 29 se limpia la señal vendedor y luego se ingresa al bloque de VALIDAR-VENDEDOR.

En el bloque VALIDAR-VENDEDOR se espera 30 segundos la respuesta del cliente (líneas 85 y 86). Si la respuesta no llega, se cancela la llamada y se mueve un 1 a fin-de-llamada.

Si la respuesta es ESC, se activa la señal de fin-de-llamada y se cancela la llamada (líneas 101 y 102). Se sale por línea 31 (la señal de vendedor continua en 0) y se saltean desde la línea 32 hasta la 37, pero se continúa dentro del bloque PROCESO_PRINCIPAL. Si el cliente ingresa el código de vendedor, se continua por línea 92, luego se analiza el código de vendedor (línea 93), si es incorrecto se solicita que lo vuelva a ingresar, se mantienen todas las señales en 0 y se espera que el cliente ingrese nuevamente el código de vendedor.

Si el cliente es válido, se continúa por línea 94 enviando al cliente mensaje que ingrese el importe de la compra, se activa la señal de vendedor y se continúa por línea 31 para ingresar al bloque de INGRESAR-IMPORTE si previamente se ha limpiado la señal de fin-importe.

En INGRESAR-IMPORTE se espera 30 segundos la respuesta del cliente (líneas 107 y 108). Si no llega respuesta o la respuesta es ESC, se activa la señal de fin-de-llamada y se cancela la llamada (líneas 109 y 110). Se sale del bloque de INGRESAR-IMPORTE se saltearán las líneas que deberían continuar si el programa estuviera completo (línea 35 y las que hicieran falta) y se inicia nuevamente el bloque PROCESO_PRINCIPAL.

Si se ingresara un importe mayor que el saldo disponible del cliente, se le enviará al cliente un mensaje solicitando que ingrese nuevamente el importe por saldo insuficiente (líneas 121 a 124). Se mantiene todas las señales en 0 para que el programa se mantenga dentro del bloque INGRESAR-IMPORTE. Si el saldo es suficiente, se resta el importe al saldo-disponible y se activa la señal de FIN-IMPORTE para salir del bloque INGRESAR-IMPORTE. Luego se continuaría por línea 35 que no ha sido documentada.

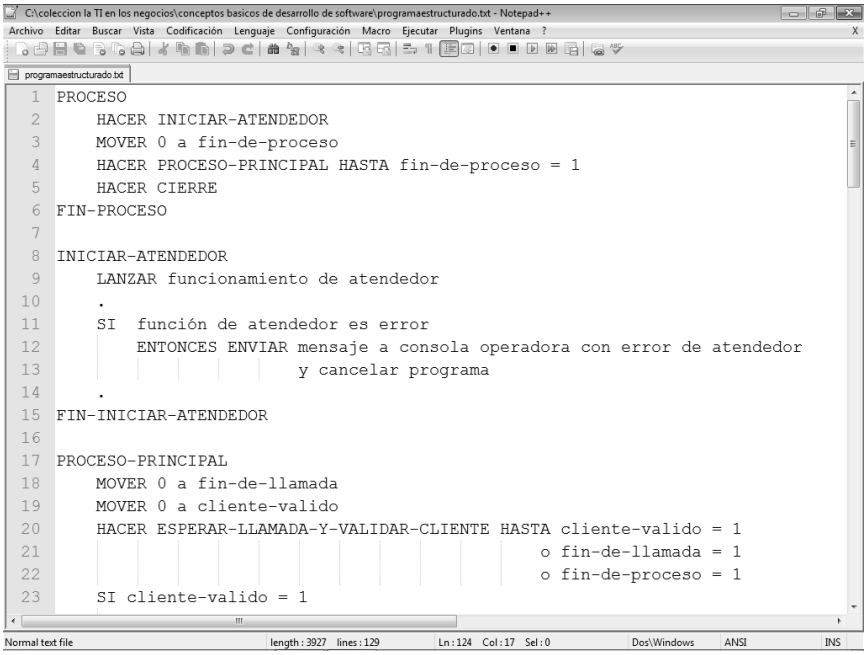


Figura 13a

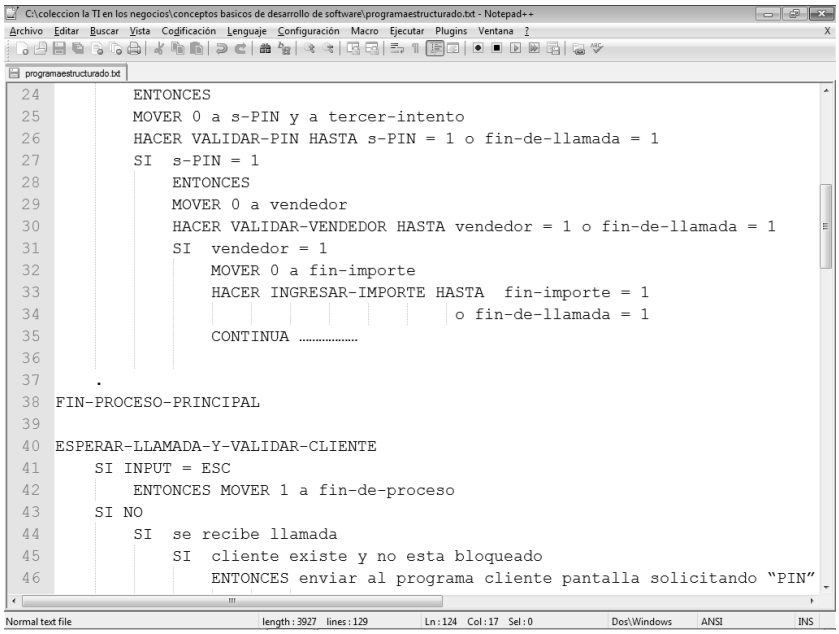


Figura 13b

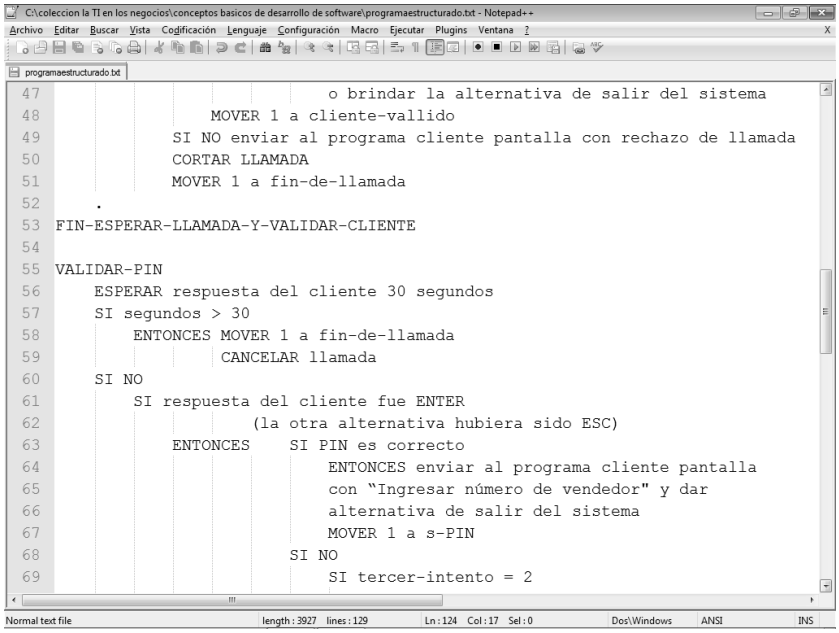


Figura 13c

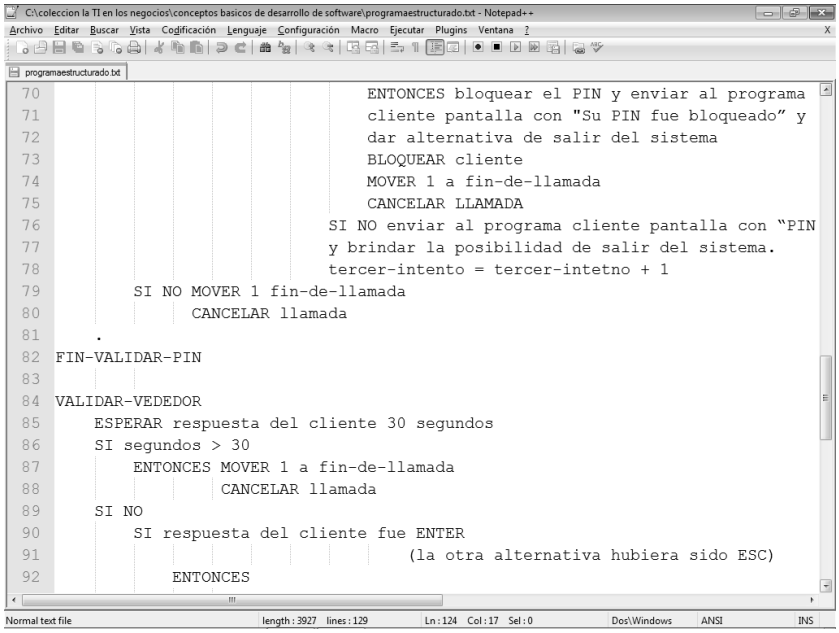


Figura 13d

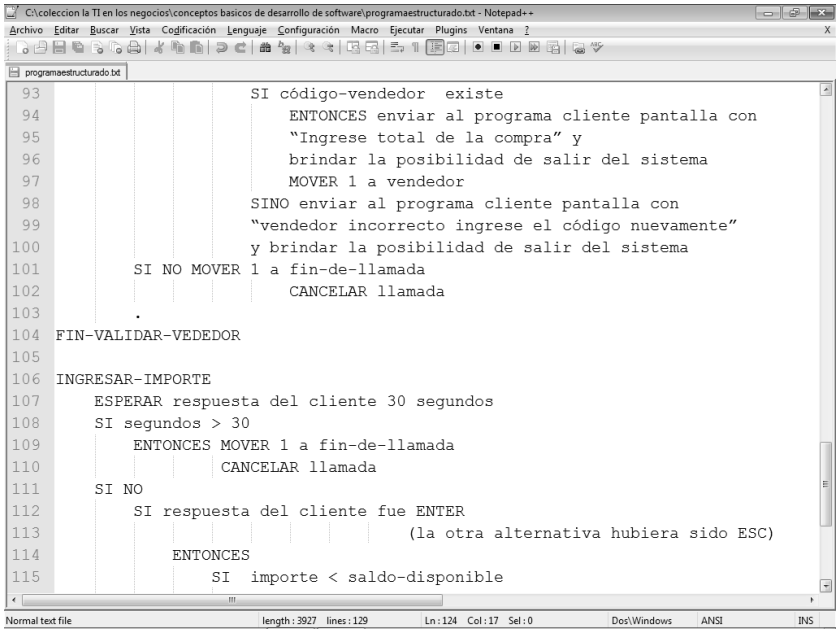


Figura 13e

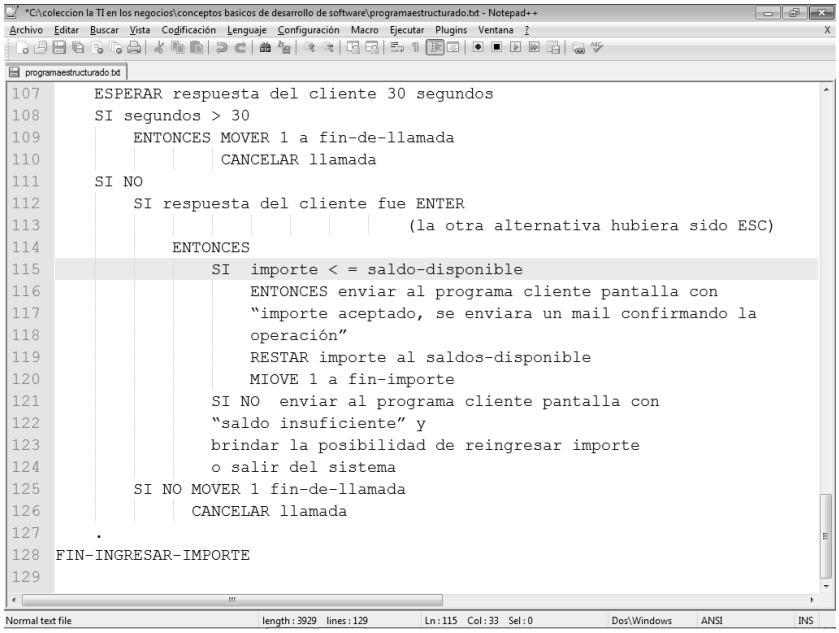


Figura 13f

6.2-Programación orientada a objeto

Tratando de ser explícito, el lector debe imaginar un programa para un robot que pueda efectuar la construcción de una pared de ladrillos. El robot procederá de la siguiente forma:

a) Comprar varias bolsas de cemento, cal y arena.
b) Comprar ladrillos.
c) Hacer mezcla para pegar ladrillos.
d) Ir armando la pared con los ladrillos y la mezcla para pegar ladrillos.
e) Hacer mezcla para revoque grueso.
f) Humedecer la pared.
g) Colocar revoque grueso.
h) Hacer mezcla para revoque fino.
i) Colocar revoque fino.

Hay que tener en cuenta que las mezclas no son iguales para pegar ladrillos que para hacer un revoque grueso o un revoque fino.

La programación estructurada tiene, básicamente, dos problemas fundamentales.

a) Como en el caso del robot construyendo la pared, el programa estructurado tiene un solo destino que es construir la pared en cuestión. Si se estudia detenidamente la secuencia del programa, se podrá observar que muchas de las instrucciones que se utilizarán podrían servir para otros programas. Pero al estar el programa armado en forma estructurada, estas instrucciones se utilizarán en esta aplicación únicamente.

b) El programa construido en forma estructurada es básicamente línea por línea desde el principio hasta el fin. Por lo tanto, es muy difícil que en un mismo programa puedan intervenir varios programadores.

La programación orientada a objetos tiene como objetivos principales primero: construir una aplicación, pero de tal manera que el código que se programe pueda ser utilizado en otras aplicaciones; y segundo: que en la construcción del programa, que en forma estructurada solo podía intervenir una persona, ahora puedan participar varias personas con total independencia una de otra.

Ahora se verá cómo agrupar funciones (instrucciones de programa) en torno de los objetos, utilizando el caso del robot construyendo la pared.

En el relato anterior, se pueden encontrar los siguientes objetos:

a) Bolsa de cemento.
b) Bolsa de cal.
c) Bolsa de arena.
d) Ladrillos.
e) Pared.
f) Mezcla.
g) Revoque fino.
h) Revoque grueso.

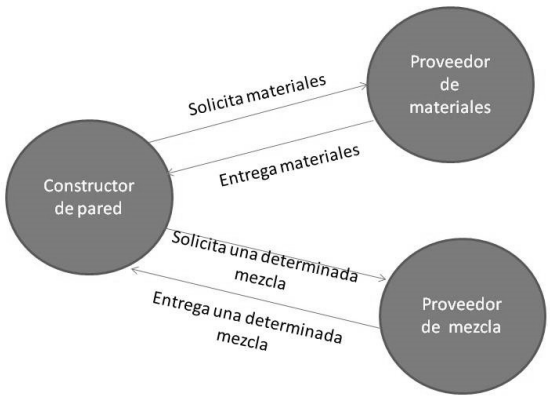
De un análisis intuitivo, se podría decir que en realidad hay menos objetos en juego. Podría decirse que los objetos (en forma genérica) son tres:

a) Materiales de construcción.
b) Pared.
c) Mezcla (para fino, grueso o pegar ladrillos).

Pensando en la posibilidad de reutilizar el código y separar el programa en varios subprogramas para que puedan intervenir varios programadores en la construcción, se podría pensar en los siguientes tres programas.

Uno que entregue materiales de construcción, otro que realice diferentes mezclas (según las necesidades) y otro que construya paredes.

De tal forma que se construyan pequeñas aplicaciones que se brindarán servicios entre ellas. En algunos casos, una aplicación hará de prestador de servicios y en otros de usuario de servicios:



Programando de esta forma, podría reutilizarse cada uno de los pequeños programas en otras aplicaciones que lo puedan requerir (por ejemplo: un constructor de veredas). Además, es absolutamente factible que en la misma aplicación trabajen tres programadores con absoluta independencia.

En la programación orientada a objetos, a cada uno de los tres programas se los llama clases. Para que las clases puedan brindar servicios, deben estar instanciadas, es decir que deben estar cargadas en memoria. Una clase instanciada es un objeto, es decir que un programa (clase) cargado en la memoria de la computadora es un objeto.

Es importante que también quede claro que una clase (programa) puede instanciar múltiples objetos a semejanza de la clase. Por ejemplo, en el caso de la construcción de la pared, se podría haber instanciado, desde la clase mezcla (instanciada), tres objetos: uno para realizar mezclas para pegar ladrillos, otra de mezcla para revoque grueso y otro de mezcla para revoque fino. Estos tres objetos se lograrían de la clase mezcla con el simple cambio de algunos valores de las variables (atributos) al instanciar la clase (Ver Figura 14). La clase mezcla instanciada se la llama constructor.

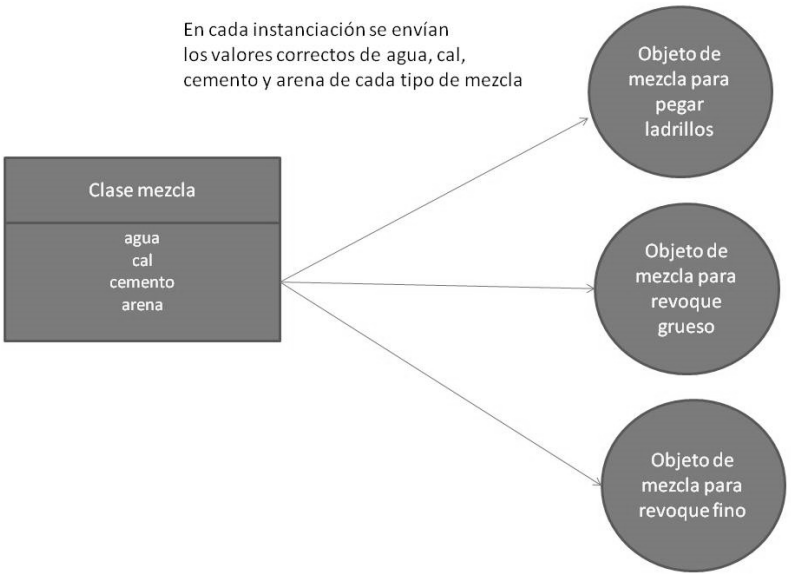


Figura 14

Ejemplo con un programa de monitoreo de motores:

Se hará un ejemplo que consiste en monitorizar tres motores a combustión simultáneamente y activar alarmas cuando las funciones de los motores salen de los parámetros esperados.

La narrativa del sistema sería la siguiente:

El sector de laminado cuenta con tres motores a explosión los cuales deben mantenerse operando siempre entre los valores preestablecidos (parámetros de los motores). El sistema deseado deberá monitorear los motores y verificar que siempre se encuentren entre los parámetros deseados, de lo contrario deberá emitir una alarma avisando que se ha producido una situación anormal en alguno de los ellos.

El proceso más simple para identificar las clases posibles a utilizar, consiste en subrayar los sustantivos en la narrativa. De lo dicho, surgen cuatro clases posibles, luego hay que realizar un análisis más profundo para decidir cuáles serán las clases definitivas. En esta obra, se siguen los patrones de diseño de clases GRASP desarrollados por Craig Larman[1]. GRASP es un acrónimo de *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades).

En función del proceso de subrayados de sustantivos en la narrativa, surgen como posibles candidatas las siguientes clases:

[1] Larman, C.: *UML y Patrones*, Pearson, 2003.

- Motores.
- Parámetros.
- Sistema de monitoreo.
- Alarma.

Los Patrones, básicos, de GRASP son:

Experto en información: la responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos).

Creador: se asigna la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando:

- B contiene a A.
- B es una agregación (o composición) de A.
- B almacena a A.
- B tiene los datos de inicialización de A (datos que requiere su constructor).
- B usa a A.

Bajo Acoplamiento: debe haber pocas dependencias entre las clases. Si todas las clases dependen de todas, ¿cuánto software se podrá extraer de un modo independiente y ser reutilizado en otro proyecto?

A eso se debe la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases.

Alta Cohesión: cada elemento de nuestro diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos. Un ejemplo de una baja cohesión son clases que hacen demasiadas cosas.

Controlador: asignar la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etcétera). El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

De lo expuesto y siguiendo los patrones GRASP, la clase Sistema de monitoreo es la gran candidata a ser controladora. La pregunta que se debe realizar ahora es: ¿qué tareas debe realizar el Sistema de monitoreo?

Tareas del sistema de monitoreo: 1) mostrar al usuario si algún motor esta fuera de su parámetro.

Qué datos necesita saber para poder cumplir con esta misión: los parámetros de cada motor y el funcionamiento real de cada motor.

En el modelo de clases, aparece la clase motores. ¿Qué responsabilidad tendrá que cumplir la clase motores?

Tareas de la clase motores: obtener los valores de funcionamiento del motor.
Qué datos necesita saber para poder cumplir con esta misión: el funcionamiento real de cada motor.

En el modelo de clases, aparece la clase parámetros. ¿Qué responsabilidad tendrá que cumplir la clase parámetros?

Tareas de la clase parámetros: obtener de un archivo los valores de los parámetros para cada motor (valores aceptables de cada motor).

Qué datos necesita saber para poder cumplir con esta misión: los parámetros de cada motor contenidos en un archivo de parámetros.

Faltaría ver qué tareas debería realizar la clase alarmas: parecería que asignarle alguna tarea a esta clase implicaría bajar la cohesión de las otras y aumentar la interdependencia entre las clases existentes. Por tal motivo, la eliminaremos de nuestro modelo de implementación.

6.2.1-Ejemplo de clases para la aplicación descripta anteriormente

Son tres clases (programas) que se inician con el nombre de la clase.

En el ejemplo, se utiliza un seudocódigo basado en Java.

1-Clase de sistema de monitoreo (Ver Figuras 15a, 15b y 15c):

La línea 1 importa un paquete de clases de manejadores de plc de motores y de las cuales se utilizarán los métodos plc.obtenerAgua(), plc.obtenerAceite() y plc.obtenerRpm().

El concepto de public de la línea 2 significa que la instanciación de esta clase (el objeto que surja de esta clase) podrá ser accedida por cualquier otro programa.

Los nombres de las clases se escriben comenzando cada palabra en mayúscula, en este caso: SistemaDeMonitoreo.

Después de doble barra (//), se indican comentarios del programa que no son instrucciones ejecutables.

Las clases contienen, básicamente, atributos y métodos. Los atributos son variables del programa, pertenecientes a las clases, que al momento de instanciarse (creación del objeto) se les asigna un valor. También, se puede decir que los atributos de una clase son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades. Los métodos son las instrucciones que por lo general permiten cambiar los valores de los atributos de los objetos u obtener información de los objetos.

En la línea 7, se abre el juego de llaves que contiene las instrucciones de la clase SistemaDeMonitoreo.

La instrucción de la línea 5 es una muy especial que hace que la clase se autoinstancie. En todas las aplicaciones, siempre habrá una de las clases que se autoinstancie y, a partir de ella, comience la cadena de instanciaciones de otras clases.

En la línea 7, se abre un juego de llaves, el lector debe fijarse que esta clase tiene tres juegos de llaves. El primer juego es prácticamente formal, ya que incluye las instrucciones de la clase. Luego, el segundo juego de la línea 7 y el tercer juego de la línea 17. El porqué de estos dos últimos juegos es porque las instrucciones encerradas en la tercera llave de la línea 17 es un bucle de *n* repeticiones que se realizarán hasta que alguien cancele el programa (según el seudocódigo de la línea 16). Este programa se ejecutará como un demonio (programa que se ejecuta continuamente en segundo plano sin control directo de un usuario) que estará permanentemente monitoreando los motores. Si no se separaban en dos juegos de instrucciones, no se podría quitar del bucle a las instrucciones entre la línea 7 y la 16.

La instrucción de la línea 8 es para instanciar la clase MotorEnFuncionamiento que luego se verá qué servicios presta en la aplicación. El objeto instanciado se llamará motor1 y el parámetro que se envía en la instanciación es 81 (que se encuentra entre paréntesis en el término derecho de la igualdad). Esta es la instrucción que java utiliza para instanciar las clases. Las instrucciones 9 y 10 son iguales, pero para el motor2 y el motor3 respectivamente. El lector podrá observar que la instanciación de cada motor utiliza el parámetro 81, 82 y 83 sucesivamente. Este último número es el parámetro que se pasa al objeto MotorEnFuncionamiento como el puerto al cual se conectará el objeto plc para obtener información de la computadora del motor correspondiente.

En la línea 1,1 se instancia la clase ParametroDeMotor cuyo objeto se denomina aplicación y no se envían parámetros en su instanciación.

En la línea 12, se instancia la clase PlcDriver cuyo objeto se denomina plc y no se envían parámetros en su instanciación.

En la línea 16, comienza el bucle (con una instrucción de español estructurado).

La línea 18 está invocando un método del objeto aplicación que le retorna el parámetro del nivel de temperatura del agua desde, del motor 1, y lo compara con el valor que le retorna un método del objeto motor1 con el valor real de la temperatura del agua del motor 1.

Para invocar un método de un objeto, se utiliza el nombre del objeto luego un punto y luego el método invocado: nombredelobjeto.metododelobjeto(). El último juego de paréntesis es para enviar parámetros al método si fuera necesario.

En la línea 19, invocando a un método del objeto aplicación que le retorna el parámetro del nivel de temperatura del agua hasta, del motor 1, y lo compara con el valor que le retorna un método del objeto motor1 con el valor real de la temperatura del agua del motor 1. Esta última invocación podría haberse evitado si en la primera oportunidad que se invoca el valor real de la temperatura del agua se hubiera guardado el dato en una variable de la clase SistemaDeMonitoreo (en el ejemplo, a los efectos de simplificación, se invoca nuevamente el método).

Si el valor del agua real obtenido del motor1 no se encuentra entre los valores de los parámetros, se desplegará en el monitor una leyenda que dirá “el motor 1 está en problemas” (línea 20).

En los siguientes dos SI (líneas 22 y 26), se realiza exactamente la misma tarea, pero en el primer caso para monitorear la temperatura del aceite y, en el segundo, para monitorear las revoluciones por minuto (rpm) del motor 1.

A partir de la línea 34, comienzan tres juegos de SI para controlar el motor2 y, a partir de la línea 50, otros tres SI para controlar el motor3.

En el punto 3.1 del Capítulo III, se podrá observar esta programación en un gráfico muy útil que se denomina Diagrama de colaboración; y en el punto 3.2 del Capítulo III, se puede observar el diagrama de secuencia.

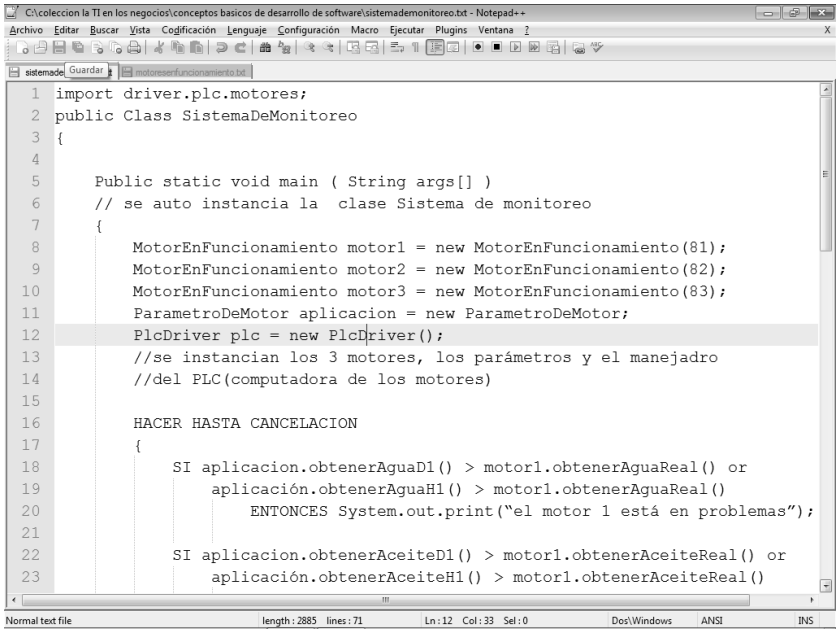


Figura 15a

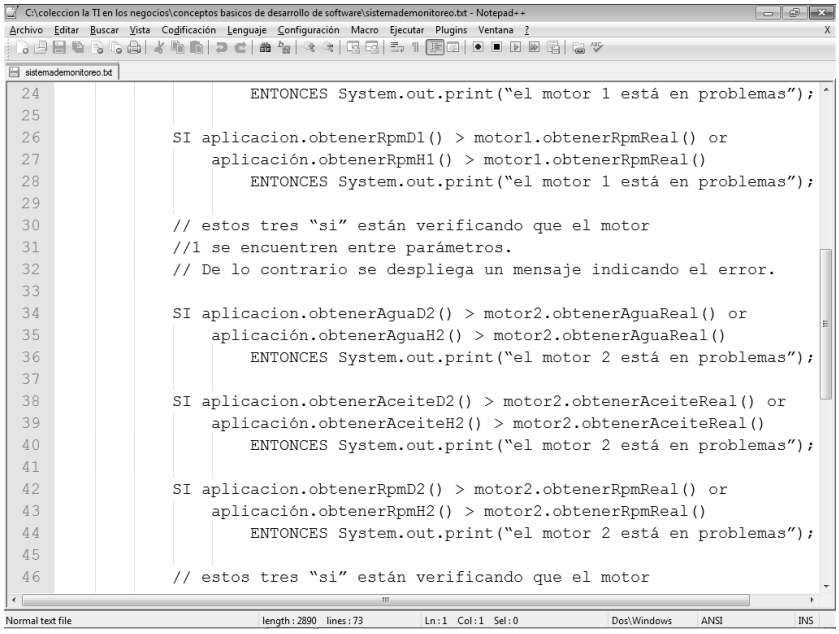


Figura 15b

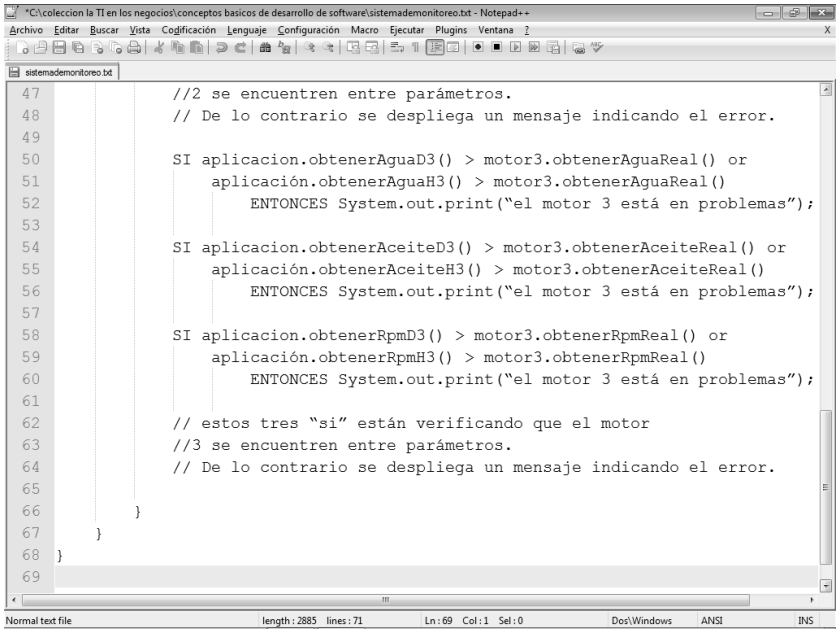


Figura 15c

2-El tratamiento de la clase Motores en funcionamiento es el siguiente (Ver Figuras 16a y 16b):

En la línea 1, se declara la clase MotorEnFuncionamiento.

De la línea 3 a la 5, se declaran como números enteros tres variables, una para almacenar el valor real del agua del motor, la otra para almacenar el valor real del aceite y la tercera para el valor real de la rpm. Cada instancia (motor1, motor2 y motor3) tendrá el valor real de cada motor en cuestión. Recuerde el lector que la clase SistemaDeMonitoreo creó una instancia para cada motor.

La línea 6 contendrá el valor del puerto serie al que se conectará el objeto plc.

Si el primer método de la clase se llama igual que la clase, a este método se lo denomina constructor y se ejecuta cuando la clase se instancia, es decir, cuando se crea el objeto. En el ejemplo, el constructor se extiende de la línea 8 a la línea 12. El constructor es un método que recibe el número de puerto en la variable s. Este número de puerto es pasado por la clase SistemaDeMonitoreo (una vez instanciada) al momento de instanciar el objeto motorX (ver las líneas 8, 9 y 10 de la clase SistemaDeMonitoreo. El constructor MotorEnFuncionamiento recibe el parámetro que lo declara como entero y se lo asigna a la variable *s*. Luego, dentro del método, en la línea 10, se le asigna *s* a la variable puertoSerial.

La línea 14, que contiene la instrucción public obtenerAguaReal(), es un método que pone a disposición de otros objetos para obtener el valor de la temperatura del agua del motor. Este dato lo logra utilizando un método del objeto plc (línea 16). El objeto plc (no documentado) se conectará al puerto con el cual se instanció la clase motor (puede ser el motor 1 o el 2 o el 3). La instrucción return de la línea 18 es el valor que devuelve el método obtenerAguaReal().

La misma funcionalidad que tiene el método obtenerAguaReal() es la que tiene los métodos obtenerAceiteReal() y obtenerRpmReal (), métodos contenidos en las líneas 21 y 28 respectivamente.

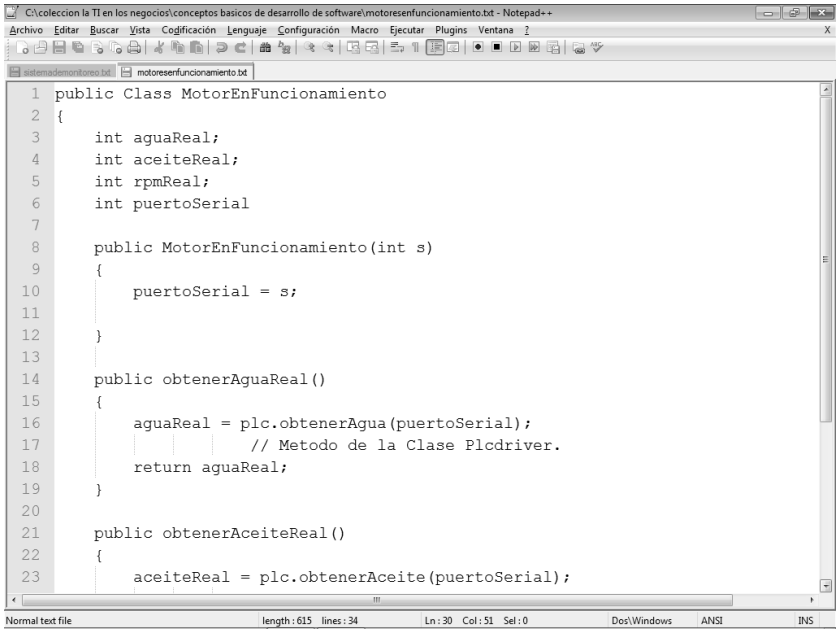

C:\coleccion la TI en los negocios\conceptos basicos de desarrollo de software\motoresenfuncionamiento.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
sistemademonitoreo.txt
motoresenfuncionamiento.txt
public Class MotorEnFuncionamiento
{
int aguaReal;
int aceiteReal;
int rpmReal;
int puertoSerial
public MotorEnFuncionamiento(int s)
{
puertoSerial = s;
}
public obtenerAguaReal()
{
aguaReal = plc.obtenerAgua(puertoSerial);
// Metodo de la Clase Plcdriver.
return aguaReal;
}
public obtenerAceiteReal()
{
aceiteReal = plc.obtenerAceite(puertoSerial);
Normal text file
length : 615 lines : 34
Ln : 30 Col : 51 Sel : 0
Dos\Windows
ANSI
INS

Figura 16a

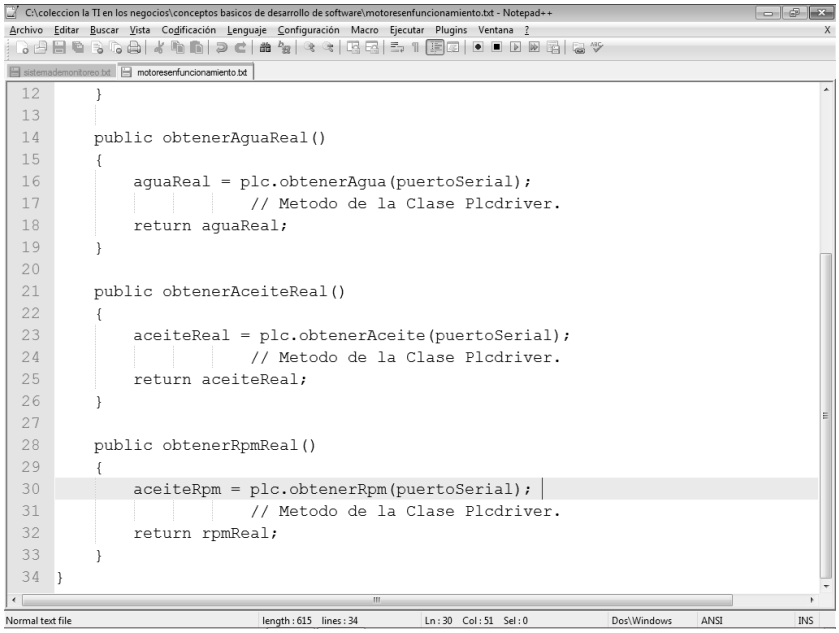

C:\coleccion la TI en los negocios\conceptos basicos de desarrollo de software\motoresenfuncionamiento.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
sistemademonitoreo.txt
motoresenfuncionamiento.txt
12 }
13
14 public obtenerAguaReal()
15 {
16 aguaReal = plc.obtenerAgua(puertoSerial);
17 // Metodo de la Clase Plcdriver.
18 return aguaReal;
19 }
20
21 public obtenerAceiteReal()
22 {
23 aceiteReal = plc.obtenerAceite(puertoSerial);
24 // Metodo de la Clase Plcdriver.
25 return aceiteReal;
26 }
27
28 public obtenerRpmReal()
29 {
30 aceiteRpm = plc.obtenerRpm(puertoSerial);
31 // Metodo de la Clase Plcdriver.
32 return rpmReal;
33 }
34 }
Normal text file
length : 615 lines : 34
Ln : 30 Col : 51 Sel : 0
Dos\Windows
ANSI
INS

Figura 16b

3-La clase ParametroDeMotores (Ver Figuras 17a, 17b, 17c, 17d, 17e, 17f, 17g y 17h):

Esta clase se encargara de leer un archivo con todos los parámetros de cada uno de los motores y los pondrá a disposición de otras clases.

El archivo que leerá tiene la siguiente estructura. Es un archivo plano con los siguientes registros:

motor 1	agua desde
motor 1	agua hasta
motor 1	aceite desde
motor 1	aceite hasta
motor 1	rpm desde
motor 1	rpm hasta
motor 2	agua desde
motor 2	agua hasta
motor 2	aceite desde
motor 2	aceite hasta
motor 2	rpm desde
motor 2	rpm hasta
motor 3	agua desde
motor 3	agua hasta
motor 3	aceite desde
motor 3	aceite hasta
motor 3	rpm desde
motor 3	rpm hasta

Este es un programa a los efectos de ejemplificar la programación orientada a objeto. Desde luego que una clase de este tipo debería haber estado orientada a una cantidad ilimitada de motores y de parámetros. Pero, haberle dado esta funcionalidad a la clase hubiera obligado una explicación más profunda de instrucciones especiales que no vienen al caso. Por otro lado, la carga de registros debió ser efectuada mediante un bucle, pero se prefirió usar una forma extremadamente explícita, aunque primaria.

La línea 1 de esta aplicación carga la clase RegistroCuenta, que no está documentada en esta explicación.

La línea 2 declara la clase ParametroDeMotores que se construirá con el método de la línea 23 (que es el constructor de esta clase) que finaliza en la línea 106.

De la línea 4 a la 21, se declaran todas las variables donde se almacenarán los valores leídos del archivo de parámetros.

El constructor de esta clase instancia la clase RegistroCuenta y denomina al objeto registro (línea 25).

Desde la línea 30, se inicia la carga de todos los registros del archivo de parámetros en las variables declaradas anteriormente. Con el método obtenerSiguienteRegistro() va levantando cada fila del objeto registro y cargándola a la variable correspondiente.

Por último, de la línea 108 a la 181, se publican los métodos para obtener los valores de cada parámetro.

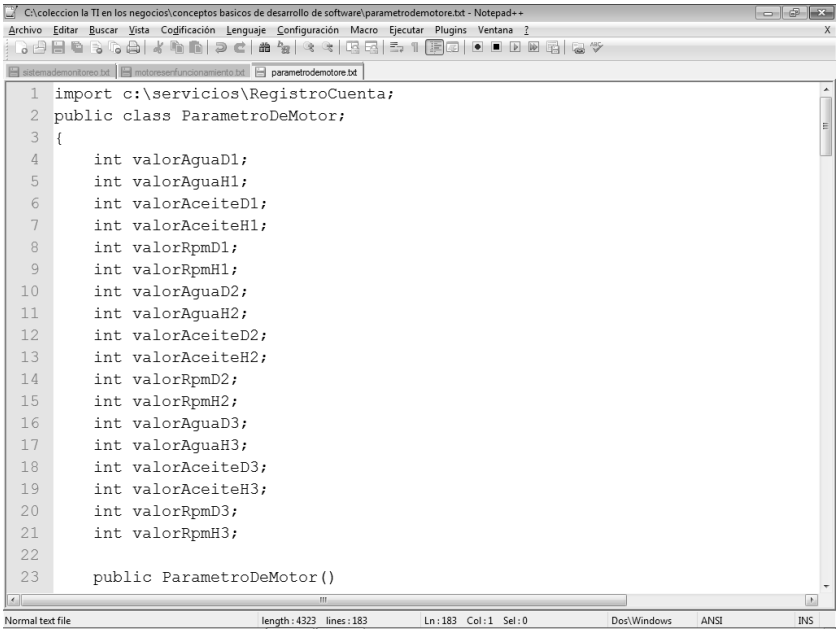


Figura 17a

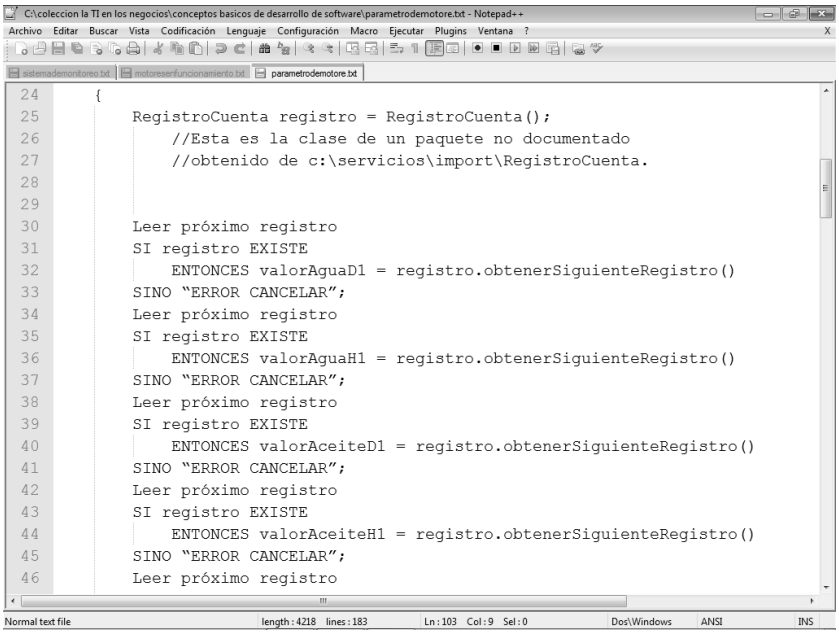


Figura 17b

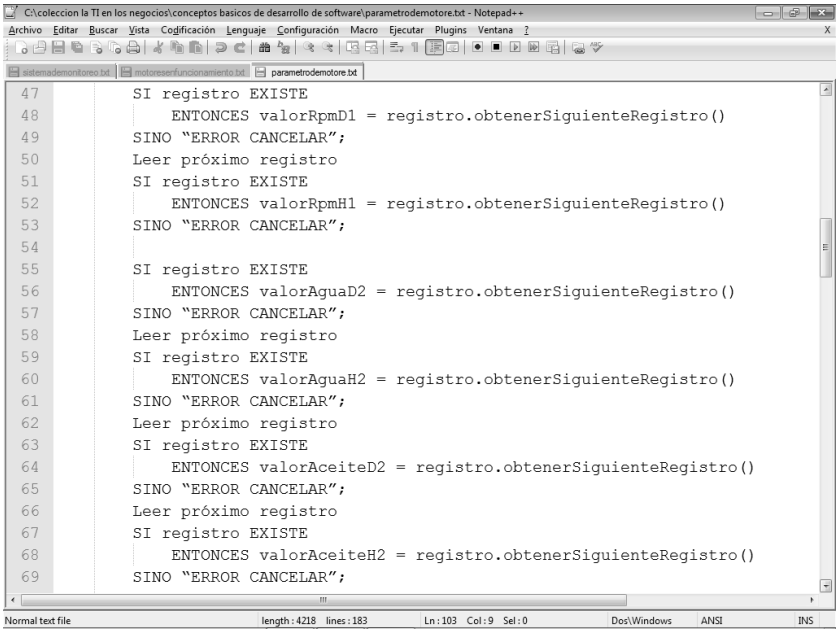

C:\coleccion la TI en los negocios\conceptos basicos de desarrollo de software\parametrodemotore.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
sistemademontoreo.txt
motoresenfuncionamiento.txt
parametrodemotore.txt
47 SI registro EXISTE
48 ENTONCES valorRpmD1 = registro.obtenerSiguienteRegistro()
49 SINO "ERROR CANCELAR";
50 Leer próximo registro
51 SI registro EXISTE
52 ENTONCES valorRpmH1 = registro.obtenerSiguienteRegistro()
53 SINO "ERROR CANCELAR";
54
55 SI registro EXISTE
56 ENTONCES valorAguaD2 = registro.obtenerSiguienteRegistro()
57 SINO "ERROR CANCELAR";
58 Leer próximo registro
59 SI registro EXISTE
60 ENTONCES valorAguaH2 = registro.obtenerSiguienteRegistro()
61 SINO "ERROR CANCELAR";
62 Leer próximo registro
63 SI registro EXISTE
64 ENTONCES valorAceiteD2 = registro.obtenerSiguienteRegistro()
65 SINO "ERROR CANCELAR";
66 Leer próximo registro
67 SI registro EXISTE
68 ENTONCES valorAceiteH2 = registro.obtenerSiguienteRegistro()
69 SINO "ERROR CANCELAR";
Normal text file length : 4218 lines : 183 Ln : 103 Col : 9 Sel : 0 Dos\Windows ANSI INS

Figura 17c

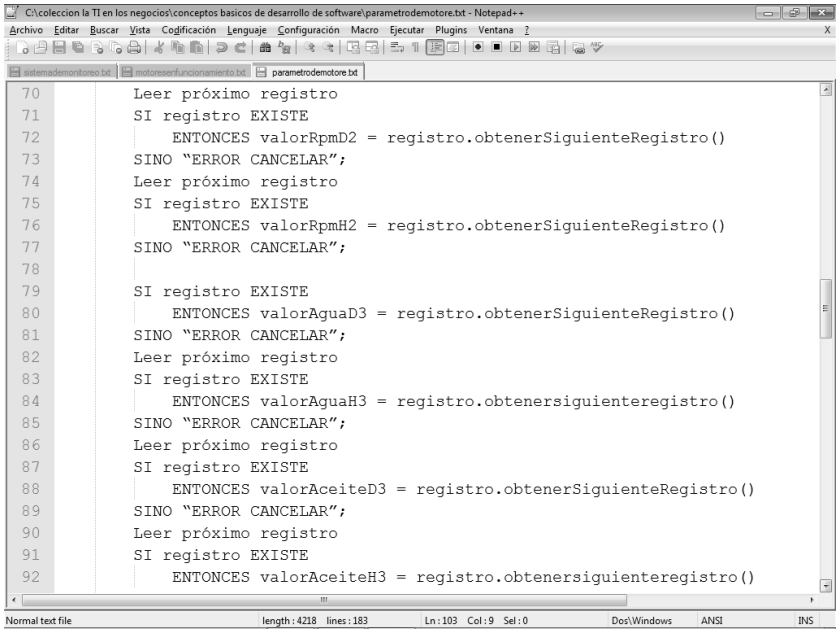

C:\coleccion la TI en los negocios\conceptos basicos de desarrollo de software\parametrodemotore.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
sistemademonitoreo.txt
motoresenfuncionamiento.txt
parametrodemotore.txt
70 Leer próximo registro
71 SI registro EXISTE
72 ENTONCES valorRpmD2 = registro.obtenerSiguienteRegistro()
73 SINO "ERROR CANCELAR";
74 Leer próximo registro
75 SI registro EXISTE
76 ENTONCES valorRpmH2 = registro.obtenerSiguienteRegistro()
77 SINO "ERROR CANCELAR";
78
79 SI registro EXISTE
80 ENTONCES valorAguaD3 = registro.obtenerSiguienteRegistro()
81 SINO "ERROR CANCELAR";
82 Leer próximo registro
83 SI registro EXISTE
84 ENTONCES valorAguaH3 = registro.obtenersiguienteregistro()
85 SINO "ERROR CANCELAR";
86 Leer próximo registro
87 SI registro EXISTE
88 ENTONCES valorAceiteD3 = registro.obtenerSiguienteRegistro()
89 SINO "ERROR CANCELAR";
90 Leer próximo registro
91 SI registro EXISTE
92 ENTONCES valorAceiteH3 = registro.obtenersiguienteregistro()
Normal text file
length : 4218 lines : 183
Ln : 103 Col : 9 Sel : 0
Dos\Windows
ANSI
INS

Figura 17d

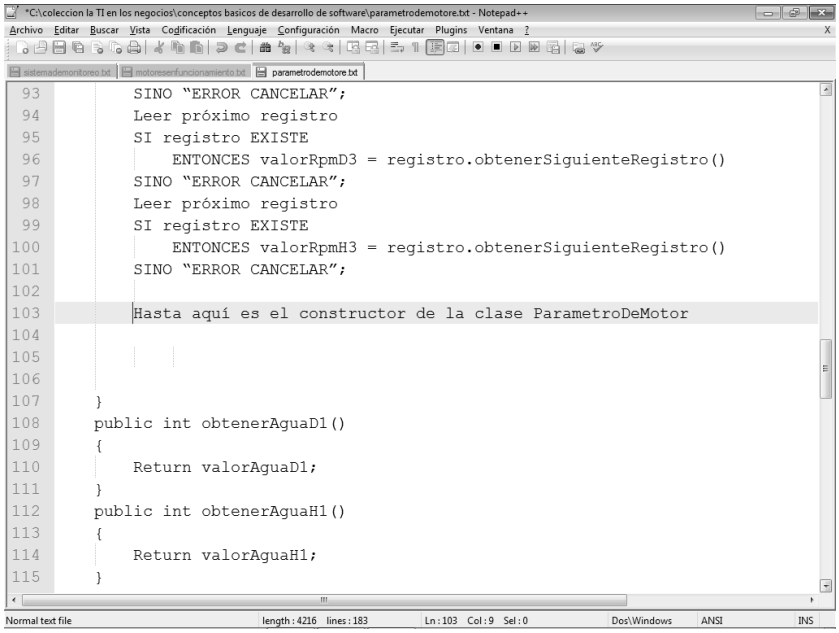

*C:\coleccion la TI en los negocios\conceptos basicos de desarrollo de software\parametrodemotore.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
sistemademonitoreo.txt
motoresenfuncionamiento.txt
parametrodemotore.txt
93 SINO "ERROR CANCELAR";
94 Leer próximo registro
95 SI registro EXISTE
96 ENTONCES valorRpmD3 = registro.obtenerSiguienteRegistro()
97 SINO "ERROR CANCELAR";
98 Leer próximo registro
99 SI registro EXISTE
100 ENTONCES valorRpmH3 = registro.obtenerSiguienteRegistro()
101 SINO "ERROR CANCELAR";
102
103 Hasta aquí es el constructor de la clase ParametroDeMotor
104
105
106
107 }
108 public int obtenerAguaD1()
109 {
110 Return valorAguaD1;
111 }
112 public int obtenerAguaH1()
113 {
114 Return valorAguaH1;
115 }
Normal text file
length : 4216 lines : 183
Ln : 103 Col : 9 Sel : 0
Dos\Windows
ANSI
INS

Figura 17e

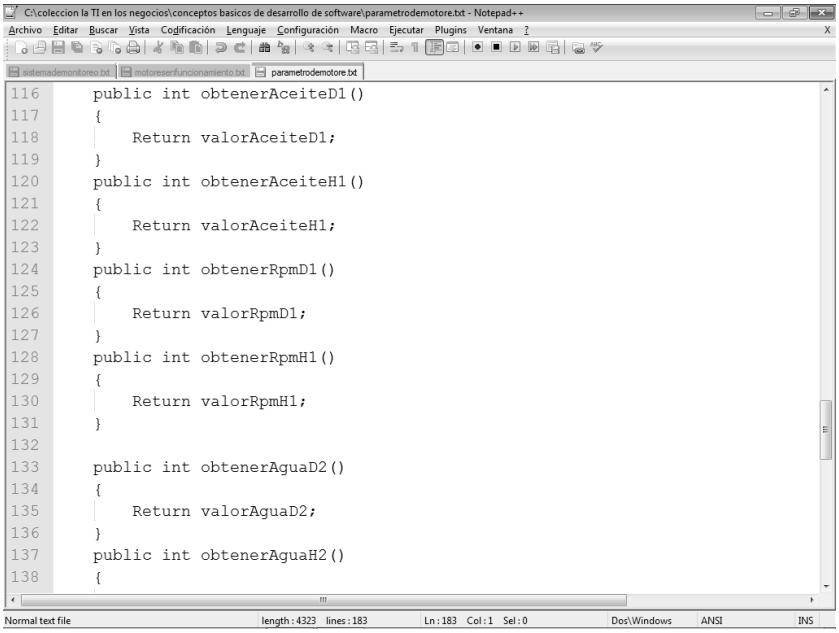

C:\coleccion la TI en los negocios\conceptos basicos de desarrollo de software\parametrodemotore.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
sistemademontoreo.txt
motoreserfuncionamiento.txt
parametrodemotore.txt
116 public int obtenerAceiteD1()
117 {
118 Return valorAceiteD1;
119 }
120 public int obtenerAceiteH1()
121 {
122 Return valorAceiteH1;
123 }
124 public int obtenerRpmD1()
125 {
126 Return valorRpmD1;
127 }
128 public int obtenerRpmH1()
129 {
130 Return valorRpmH1;
131 }
132
133 public int obtenerAguaD2()
134 {
135 Return valorAguaD2;
136 }
137 public int obtenerAguaH2()
138 {
Normal text file
length : 4323 lines : 183
Ln : 183 Col : 1 Sel : 0
Dos\Windows
ANSI
INS

Figura 17f

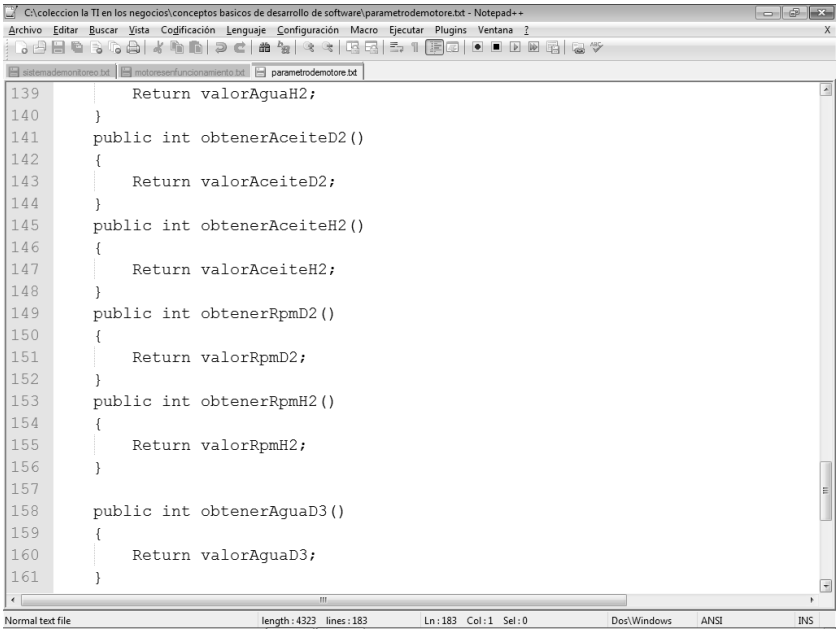

C:\coleccion la TI en los negocios\conceptos basicos de desarrollo de software\parametrodemotore.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
sistemademontoreo.txt
motoresenfuncionamiento.txt
parametrodemotore.txt
139 Return valorAguaH2;
140 }
141 public int obtenerAceiteD2()
142 {
143 Return valorAceiteD2;
144 }
145 public int obtenerAceiteH2()
146 {
147 Return valorAceiteH2;
148 }
149 public int obtenerRpmD2()
150 {
151 Return valorRpmD2;
152 }
153 public int obtenerRpmH2()
154 {
155 Return valorRpmH2;
156 }
157
158 public int obtenerAguaD3()
159 {
160 Return valorAguaD3;
161 }
Normal text file
length : 4323 lines : 183
Ln : 183 Col : 1 Sel : 0
Dos\Windows
ANSI
INS

Figura 17g

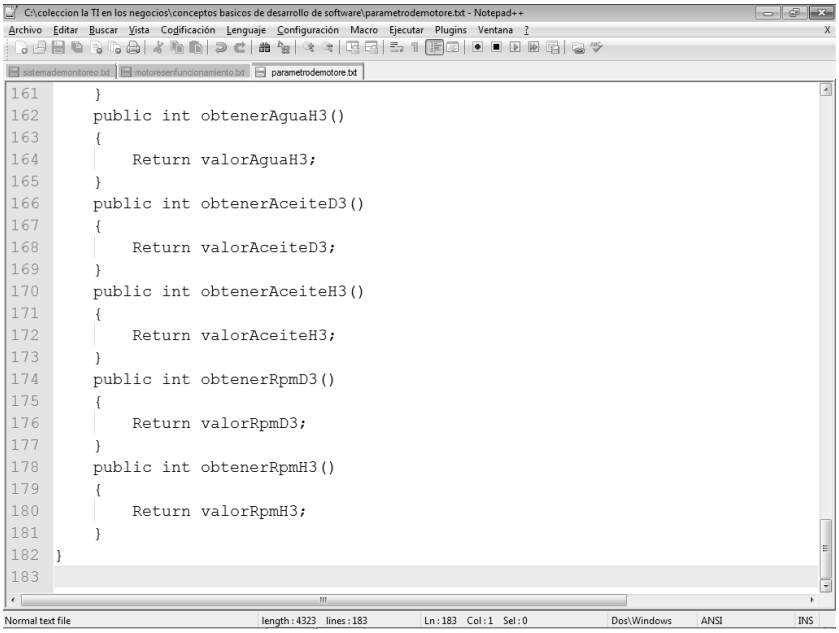


Figura 17h

6.2.1-Herencia y polimorfismo

Cuando se habla de programación orientada a objetos, no se debe dejar de nombrar dos conceptos muy importantes: herencia y polimorfismo.

Herencia: en orientación a objetos, la herencia es el mecanismo más utilizado para alcanzar uno de los objetivos más preciados en el desarrollo de software, como lo es la reutilización. A través de ella, los diseñadores pueden crear nuevas clases partiendo de una clase o de una jerarquía de clases preexistente (ya comprobadas y verificadas), evitando con ello el rediseño, la modificación y verificación de la parte ya implementada. La herencia facilita la creación de clases (a las que se denominan subclases) a partir de otros ya existentes. Esto implica la existencia de una jerarquía de clases. Una subclase obtiene todo el comportamiento (métodos) y eventualmente los atributos (variables) de la clase de la cual hereda (se la suele denominar superclase). Este mecanismo se implementa con instrucciones de programación muy simples.

Por ejemplo, una clase automotor puede ceder muchas instrucciones a otras clases (o mejor dicho subclases), como la clase camión, la clase camioneta, la clase utilitario, la clase sedan 4 puertas, etcétera.

Polimorfismo: se refiere a la posibilidad de enviar un mensaje a un grupo de objetos cuya naturaleza puede ser heterogénea. El único requisito que deben cumplir los objetos que reciben el mensaje es saber responderlo. El polimorfismo debe verse como una forma flexible de usar un grupo de objetos como si fueran uno solo. La forma de programar el polimorfismo difiere bastante de un lenguaje a otro. Hay lenguajes que requieren que los objetos polimórficos se encuentren integrados dentro de una jerarquía de objetos. Es decir que las clases que crean esos objetos deben pertenecer a una jerarquía de clases (subclase y superclase).

Ahora se realizará un programa en seudocódigo (mezcla de programa en español estructurado y de lenguaje Java) un poco más complejo a los efectos de que se pueda entender qué es lo que se busca en el modelado basado en clases.

6.3-Modelos del dominio (o modelo conceptual)

El modelo del dominio consiste en representar gráficamente las clases potenciales de la aplicación a construir (potenciales porque recién se definirá en la etapa de diseño las clases que quedarán definitivamente en la aplicación a construir), los atributos que contendrán dichas clases y las relaciones entre ellas.

6.3.1-El modelo de dominio en forma visual

Aquí se desarrollará una gráfica que mostrará las clases conceptuales, sus atributos y la relación entre ellas para el caso desarrollados del Sistema de Monitoreo.

Por el orden explicativo que se ha desarrollado, el lector puede llegar a confundirse, por eso es importante exponer nuevamente los pasos. El modelado del dominio debe realizarse previo a cualquier diseño de clases, como el que hemos realizados en el ejemplo del Sistema de Monitoreo. Aquí se ha decidido mostrar primero el diseño de clases para que el lector pueda entender de qué se está hablando cuando se mencionan clases u objetos. El modelo del dominio es un primer acercamiento a lo que en el futuro pasará a ser el diseño del sistema. Pero, era necesario explicar a que se pretende acercar con el modelo del dominio para que se entienda la idea.

Entonces, para el primer acercamiento a las potenciales clases de la aplicación a construir, lo mejor es comenzar con la narrativa de la aplicación, tal como se hizo en el caso de monitoreo de motores que expondremos nuevamente:

El sector de laminado cuenta con tres motores a explosión los cuales deben mantenerse operando siempre entre los valores preestablecidos (parámetros de los motores). El sistema deseado deberá monitorear los motores y verificar que siempre se encuentren entre los parámetros deseados, de lo contrario deberá emitir una alarma avisando que se ha producido una situación anormal en alguno de los motores.

En esta narrativa, ya se han subrayado los sustantivos y estos serán las potenciales clases a desarrollar. Lo patrones GRASP que fueron desarrollados anteriormente no son para utilizar en el modelo del dominio, sino que son para la etapa posterior de diseño de clases, cuyo objetivo es depurar las clases del modelo del dominio para llegar al diseño final.

En la figura 18, se muestra una gráfica con las clases sus atributos y las asociaciones.

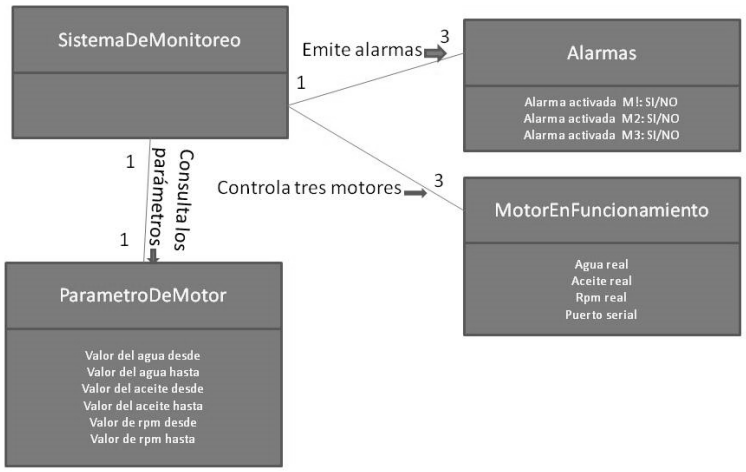


Figura 18

Cada clase puede definirse como una plantilla (una especie de formulario en blanco) que describirá las características y el comportamiento de un objeto cuando la clase se instancie en la computadora. Por ejemplo: la clase automóvil podría considerarse como un formulario a completar con los datos sobre la marca, el fabricante, el color, las dimensiones, si tienen dos, cuatro o cinco puertas, la potencia, si utiliza con combustible nafta o gasoil, etcétera. El comportamiento (o métodos) se refiere a la posibilidad de desplazarse por una carretera, frenar, acelerar, cambiar de marcha, girar, etcétera.

Cuando el formulario (la clase) se completa con los datos y se instancia en la memoria, pasa a representar un objeto en concreto que, en definitiva, es una instancia de la clase.

Una clase es, por tanto, una plantilla implementada en software que describe un conjunto de objetos con atributos y comportamiento similares.

A continuación, se realiza una enumeración, no taxativa, de categorías de clases con ejemplos.

Categoría de clases conceptuales	Ejemplos
Objetos tangibles o físicos	Registros, aviones, autos
Especificación, diseños o descripciones de las cosas	Especificación de un producto, descripción de un vuelo, descripción de una casa
Lugares	Local, fabricas, sucursales
Transacciones	Ventas, pagos, cobranzas
Líneas de transacciones	Ventas, pagos, depósitos
Roles de la gente	Cajero, jefe de ventas
Contenedores de otras cosas	Local, avión, edificio
Cosas en un contenedor	Artículos, pasajeros, líneas
Otros sistemas informáticos o electromecánicos	
Organizaciones	Departamento de ventas, compañía de alquiler de autos
Hechos	Venta, pago, reunió, vuelo, cocimiento
Procesos	Reserva de asiento, venta de un producto
Reglas y políticas	Política de reintegro, autorización de crédito
Catálogo	De producto, de servicios
Registros	Mayor, diario, verificación
Instrumentos y servicios financieros	Línea de crédito, stock, saldo
Manuales, documentos, artículos de referencia, libros	Manual de reparación, lista de precios

Las asociaciones son conexiones que revelan una relación entre las clases.

La siguiente es una lista de las asociaciones más comunes:

- A es una parte física de B.
- A es una parte lógica de B.
- A está contenida físicamente en B.
- A está contenida lógicamente en B.
- A es una descripción de B.
- A es una línea de una transacción de B.
- A se conoce/registra/recoge/informa/captura en B.
- A es miembro de B.
- A es una subunidad organizativa de B.
- A utiliza o gestiona B.
- A se comunica con B.
- A está relacionado con una transacción B.
- A es propiedad de B.
- A es un evento relacionado con B.

Las asociaciones están compuestas por:

- Nombre
- Expresión de multiplicidad
  - * cero a muchos
  - 1…* uno a muchos
  - 1…40 de uno a cuarenta
  - 5 exactamente 5
  - 3,5,8 exactamente 3 o 5 u 8

Capítulo III
El diseño

Según McGlaughlin[2], un buen diseño de software debe contemplar las siguientes tres características:

a) Debe implementar todos los requisitos explícitos contenidos en el modelo de análisis.
b) Debe ser una guía legible y comprensible para quien genere el código.
c) Debe proporcionar una imagen completa del software a programar.

1-Calidad de software

En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del sistema. Hewlett-Packard ha desarrollado un conjunto de factores de calidad del software al que se le ha dado el acrónimo de FURPS (*Functionality, Usability, Reliability, Rerformance, Supportability*): Funcionalidad, Usabilidad, Fiabilidad, Rendimiento y Capacidad de Soporte. A continuación, se definen los atributos contemplados en cada uno de estos cinco factores:

- ✓ Funcionalidad. Se valora evaluando el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global.
- ✓ Usabilidad. Se valora considerando factores humanos, la estética, la consistencia y la documentación general.
- ✓ Fiabilidad. Se evalúa midiendo la frecuencia y gravedad de los fallos, la exactitud de las salidas (resultados), el tiempo medio de fallos, la capacidad de recuperación de un fallo y la capacidad de predicción del programa.
- ✓ Rendimiento. Se mide por la velocidad de procesamiento, el tiempo de respuesta, consumo de recursos, rendimiento efectivo total y eficacia.
- ✓ Capacidad de Soporte. Combina la capacidad de ampliar el programa (extensibilidad), adaptabilidad y servicios (estos términos se resumen en el concepto de mantenimiento), así como capacidad para hacer pruebas, compatibilidad, capacidad de configuración del software, la facilidad de instalación de un sistema y la facilidad con que se pueden localizar los problemas.

2-La arquitectura de software

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Es un conjunto de modelos coherentes que proporcionan una guía para la construcción del software para un sistema de información.

3-El diseño orientado al flujo

El tipo de flujo de información es el que determina cómo se realiza la conversión del DFD a la estructura del programa. Este flujo de información puede ser de: transformación o de transacción. Un mismo DFD puede tener partes donde el flujo de información es de transformación y otra parte donde el flujo es de transacción.

El flujo de transformación: es una forma de describir a los procesos simples donde hay una clara organización en tres conjuntos de instrucciones. El primer conjunto procesa los elementos de entrada (es decir que transforma los datos de entrada en datos posibles de ser procesados (por ejemplo, los datos ingresados desde un teclado o los tonos de un teléfono, un mensaje de voz, etcétera). El segundo conjunto transforma los datos y simultáneamente los va direccionado hacia la salida del software, donde el tercer conjunto de instrucciones transforma los datos de internos a externos.

En la figura 19, se ve un flujo de transformación. La gráfica superior de la figura 19 muestra la estructura del programa (estructura en bloques) y la figura inferior muestra las burbujas asociadas a cada bloque del programa.

[2] McGlaughlin, R., *Some notes on Program Design*, Software Engineering Notes, Vol. 16, N.º 4, octubre 1991.

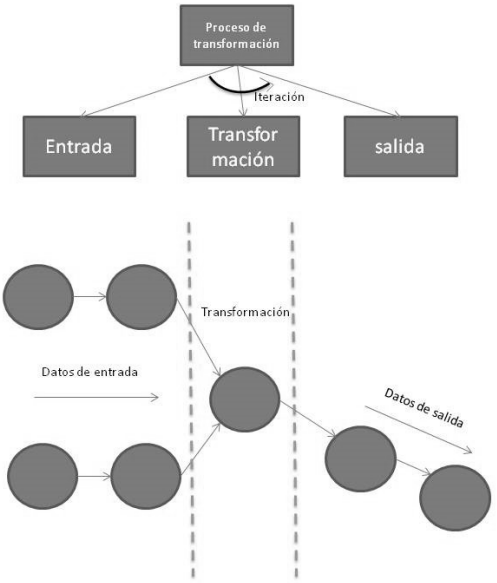


Figura 19

El flujo de transacción: se identifica por el movimiento de datos a través de un camino de llegada, donde un centro de transacción los evalúa y, de acuerdo con el valor de la comparación, el flujo sigue por uno de tantos caminos de acción (Ver Figura 20).

El flujo de transacción es mucho más simple de identificar que el de transformación, ya que el de transformación requiere identificar con más dificultad todas las subrutinas que pueden estar involucradas dentro del proceso de transformación para recién dirigirse hacia el camino de salida. Por ejemplo, el siguiente diagrama (Ver Figura 21) muestra que 1 y 2 son parte del flujo de entrada, T1 y T2 son rutinas de la transformación y el resto son salida.

Un ejemplo podría ser que un auto se para delante de una barrera, una seria de unidades de input identifican el auto y los datos de entrada pasan a un análisis y transformación para que las salidas puedan operar. Por ejemplo, que una alarma suene o que un ascensor se eleve, o que carteleras anuncien la llegada del vehículo, etcétera.

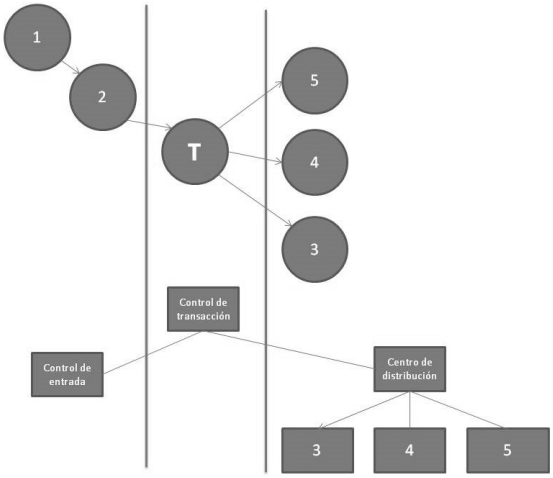


Figura 20

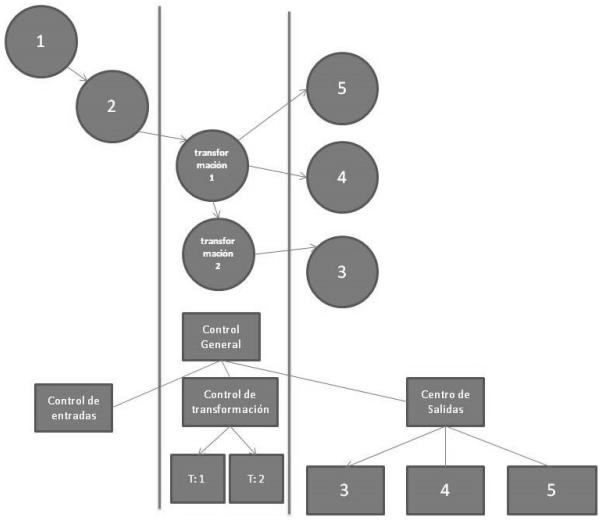


Figura 21

Si se observa el ejemplo del programa estructurado basado en el DFD "Compra por celular", se podrá observar que el DFD es claramente de transformación que queda estructurado de la siguiente forma (Ver Figura 22).

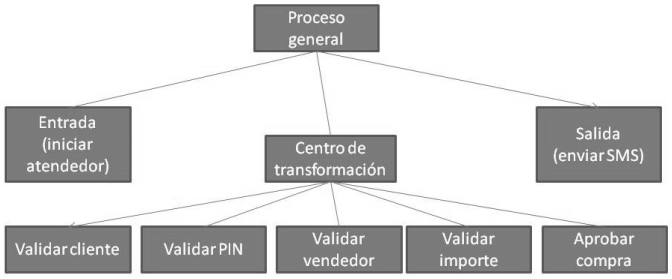


Figura 22

4-El diseño orientado a objeto

El diseño orientado a objetos es una fase de la metodología orientada a objetos para el desarrollo de software. Su uso induce a los programadores a pensar en términos de clases de software y dejando atrás el modelo de dominio.

Particularmente, durante la etapa de diseño, se efectuarán decisiones relativas a qué métodos se requieren y a qué clase se asignara cada método. Y, además, se deberá definir cómo interactuarán estos métodos, actividad nada trivial y muy importante.

Para poder diseñar esta interacción, existen una serie de diagramas que a continuación se describirán.

4.1-Diagrama de colaboración

Un diagrama de colaboración es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles. Muestran explícitamente las relaciones de los roles. Muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común.

Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Dicha implementación es llamada «enlace».

Se realizará el diagrama de colaboración para el ejemplo con el programa de monitoreo de motores de la sección 6.2 del Capítulo II (Ver Figura 23). Se debe tener en cuenta que el objeto «motores» se tomó uno solo como genérico y se evitó diagramar los tres motores para simplificar. También, los mensaje hacia el objeto aplicación se sintetizaron en una X para cada motor, de lo contrario habría que haber diagramado 12 juego de mensajes más.

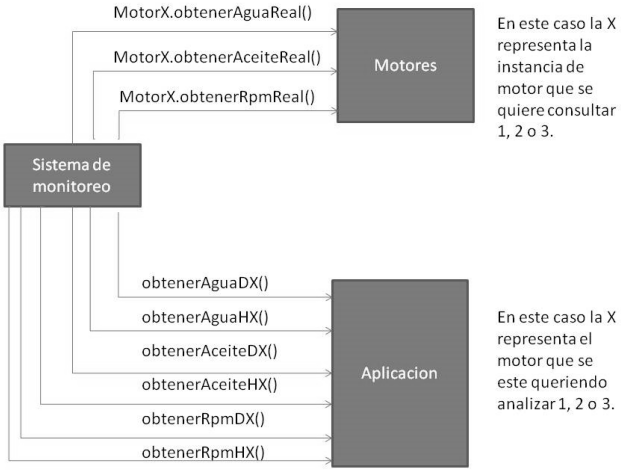


Figura 23

4.2-Diagrama de secuencia

El diagrama de secuencia va estableciendo el orden en el que se crean los objetos (cada nuevo objeto se agrega a la derecha) y la interacción entre ellos.

El ejemplo para nuestro caso de «motores» se ve en las figuras 24, 25 y 26. En este diagrama, no se simplificaron los objetos como en el caso anterior, porque se hubiera perdido el sentido de la secuencia de creación de objetos y de mensajes.

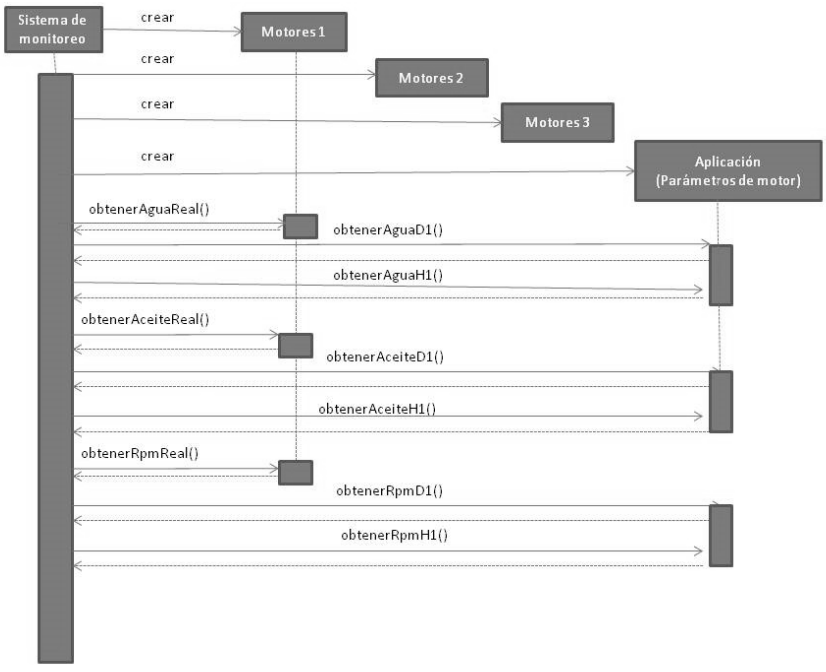


Figura 24

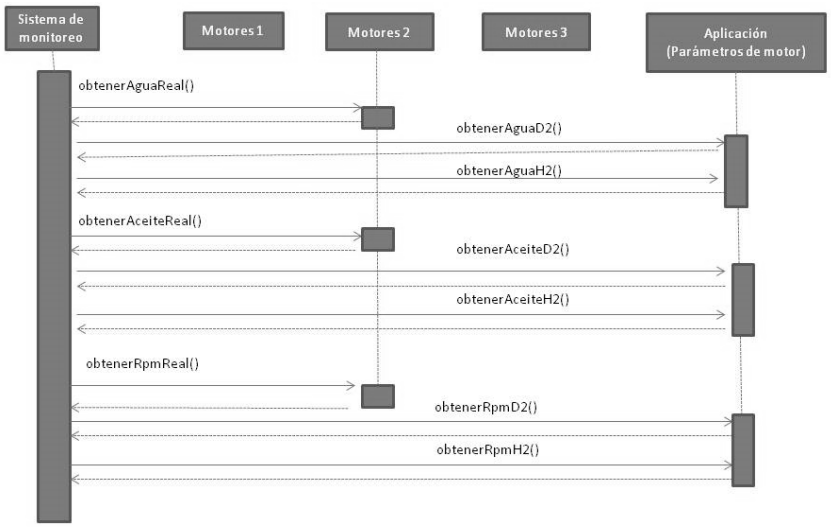

Sistema de monitoreo
Motores 1
Motores 2
Motores 3
Aplicación (Parámetros de motor)
obtenerAguaReal()
obtenerAguaD2()
obtenerAguaH2()
obtenerAceiteReal()
obtenerAceiteD2()
obtenerAceiteH2()
obtenerRpmReal()
obtenerRpmD2()
obtenerRpmH2()

Figura 25

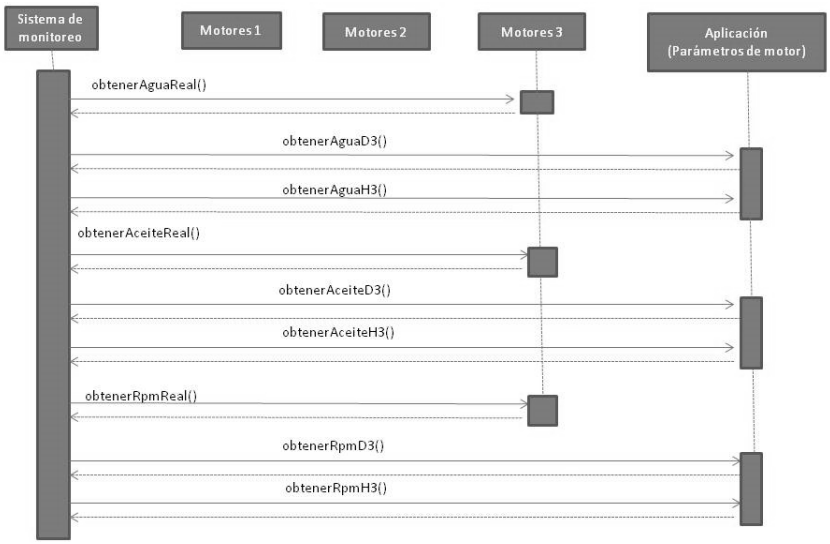

Sistema de monitoreo
Motores 1
Motores 2
Motores 3
Aplicación (Parámetros de motor)
obtenerAguaReal()
obtenerAguaD3()
obtenerAguaH3()
obtenerAceiteReal()
obtenerAceiteD3()
obtenerAceiteH3()
obtenerRpmReal()
obtenerRpmD3()
obtenerRpmH3()

Figura 26

Capítulo IV
Metodologías rápidas

Rapid Application Development (RAD) se denomina al desarrollo rápido de aplicaciones. Es una metodología de desarrollo de software, que implica el desarrollo iterativo y la construcción de prototipos. Sus principios básicos:

- Está pensado para un rápido desarrollo y una alta calidad en la construcción de software.
- Intenta reducir el riesgo inherente del proyecto partiéndolo en segmentos más pequeños para proporcionar más facilidad de cambio durante el proceso de desarrollo.
- Está orientado a producir sistemas mediante el uso de iteración y prototipos (en cualquier etapa de desarrollo).
- Promueve la participación de los usuarios y el uso de herramientas de desarrollo computarizadas, como por ejemplo los constructores de Interfaz gráfica de usuario (GUI) o las herramientas CASE (*Computer Aided Software Engineering*).
- El control de proyecto implica el desarrollo de prioridades y la definición de los plazos de entrega. Si el proyecto empieza a aplazarse, se hace hincapié en la reducción de requisitos para el ajuste, no en el aumento de la fecha límite.
- Es fundamental que los usuarios estén intensamente participando del diseño del sistema, ya sea a través de la creación de consenso estructurado en talleres o por vía electrónica.
- Produce la documentación necesaria para facilitar el futuro desarrollo y mantenimiento.

El surgimiento de RAD fue debido a la crítica constante que han sufrido el desarrollo de software siguiendo las metodologías estructuradas, como por ejemplo la metodología de Yourdon, o las orientadas a objetos, como por ejemplo RUP (*Rational Unified Process* desarrollado por la empresa Rational Software, actualmente propiedad de IBM). Estas metodologías gozan de una mala fama por los grandes periodos de tiempo que demandan para la construcción de software. El enfoque tradicional presenta ventajas y desventajas. La principal ventaja es su gran raciocinio. Primero piensa, escribe mucho, acuerda con los usuarios, firma contratos, formaliza todo, prueba todo, sigue el plan paso a paso y escribe los desvíos y sus consecuencias. La gran desventaja es que participan seres humanos. Se firma sin leer, se aprueba pensando las cosas con una visión incompleta, existen grandes diferencias entre lo que los usuarios creen decir y lo que los desarrolladores creen entender. Los distintos modelos mentales de los interlocutores llevan a grandes confusiones. Las diferencias se acumulan a lo largo del tiempo, y cuando se manifiestan, ya es tarde para corregirlas.

Los precursores de la filosofía RAD escribieron un manifiesto donde se destacan las diferencias con las metodologías hasta ese momento denominadas tradicionales (hoy se las denomina duras):

- RAD valora los individuos y su interacción sobre los procesos y las herramientas.
- En RAD el software en funcionamiento es mucho más importante que la documentación extensa.
- RAD aprecia la colaboración del cliente sobre la negociación del contrato.
- En RAD la respuesta al cambio es considerablemente mejor que el seguimiento de planes.

La alta competencia que requiere el mantenerse en el mercado necesita de una permanente actualización al cambio. El vertiginoso cambio y la alta competencia es uno de los principios que rige la economía mundial globalizada. La necesidad de hacer rápidamente obsoleto los productos y servicios requiere de altísima capacidad de adaptación. Los cambios son forzados por las necesidades de los clientes y no por los planes programados de las empresas.

Todo lo dicho requiere que el desarrollo de software sea lo vertiginoso que es el mercado y la competencia. En estos principios se asienta RAD. Las metodologías rápidas que se han destacado más son: Extreme Programming(XP), cuyo padre fundador es Kent Beck, y SCRUM desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. En esta obra, se verán los fundamentos de SCRUM, ya que es una metodología sólida dentro de las ágiles.

1-SCRUM

SCRUM es un marco de trabajo para la gestión y el desarrollo de software basada en un proceso iterativo e incremental. Los desarrolladores de software, que trabajaron en los primeros años del nacimiento del software, suelen decir que las metodologías ágiles se parece mucho a lo que hacían ellos en su época.

SCRUM nace a partir del movimiento holístico desarrollado en 1986 por Hirotaka Takeuchi e Ikujiro Nonaka. Ellos describieron una nueva aproximación holística que incrementa la rapidez y la flexibilidad en el desarrollo de nuevos productos comerciales. Un artículo de la revista *Harvard Business Review* sobre prácticas asociadas con grupos exitosos de desarrollo de producto introdujo el término «Scrum».

El modelo holístico de Hirotaka Takeuchi e Ikujiro Nonaka se denomina SECI y está formado por 4 modelos de interacción del conocimiento hablado.

- Tácito a tácito (socialización): esta dimensión explica la interacción social como la transferencia de conocimiento tácito a tácito; compartir conocimiento tácito a través de cara a cara o compartir conocimientos a través de experiencias. Por ejemplo, reuniones y tormenta de ideas pueden apoyar este tipo de interacción, ya que el conocimiento tácito es difícil de formalizar y, a menudo, requiere de tiempo y espacio específico, ya que el conocimiento tácito puede adquirirse solo a través de la experiencia compartida, como pasar tiempo juntos o viviendo en el mismo entorno. La socialización se produce normalmente en un aprendizaje tradicional, donde los aprendices aprenden el conocimiento tácito en su oficio a través de la experiencia práctica, en lugar de manuales escritos o libros de texto

- Tácito a explícito (externalización): entre el pasaje del conocimiento tácito al explícito se produce la externalización (publicación, articulación de conocimientos), desarrollo de elementos que incrustan el conocimiento tácito y que combinados permiten su comunicación. Por ejemplo, conceptos, imágenes y documentos escritos pueden apoyar este tipo de interacción. Cuando el conocimiento tácito se hace explícito, se cristaliza el conocimiento, permitiendo que sea compartida por otros y que se convierta en la base de nuevos conocimientos. Creación del concepto de desarrollo de nuevos productos es un ejemplo de este proceso de conversión.

- Explícito a explícito (combinación): combinando diferentes tipos de conocimiento explícito. Por ejemplo, mediante la combinación de las redes de comunicación y las bases de datos a gran escala. El uso creativo de las diferentes fuentes de conocimiento explícito puede soportar este modo de conversión de conocimiento. El conocimiento explícito está dentro o fuera de la organización, luego se editan, combinan y transforman en nuevas formas de conocimientos. El nuevo conocimiento explícito, a continuación, se difunde entre los miembros de la organización.

- Explícito a tácito (internalización): explícito a tácito es el proceso de internalización, incorporando el aprendizaje desde el hacer. Por otro lado, el conocimiento explícito se convierte en parte del conocimiento de un individuo y este formará parte del activo de una organización. La internalización es también un proceso de continua reflexión individual y colectiva. Es la capacidad de ver las conexiones y reconocer patrones para dar sentido a ideas y conceptos.

1.1-Características de SCRUM

SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en SCRUM son el Scrum Manager, que mantiene los procesos y trabaja de forma similar al director de proyecto; el Propietario del Producto, que representa a los *stakeholders* (interesados externos o internos); y el Team, que incluye a los desarrolladores (Ver Figuras 27 y 28).

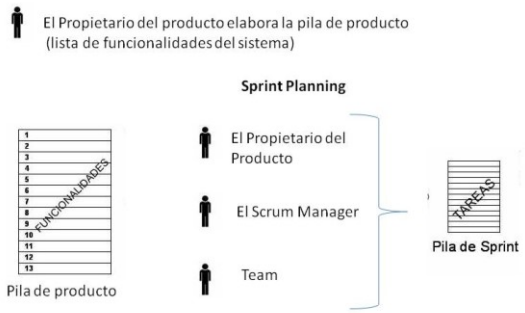


Figura 27

Figura 28

Durante cada sprint (un periodo entre una y cuatro semanas, la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable). Los sprints nunca se alargan, se limitan en el tiempo. El entregable debe estar probado para ser mostrado a los usuarios y puesto en funcionamiento. De la interacción con el usuario, pueden surgir modificaciones o mejoras que se incorporarán en el próximo sprint. El conjunto de características que forma parte de cada sprint viene de la Pila de Producto (lista de funcionalidades), que es un conjunto de requisitos de alto nivel que definen el trabajo total a realizar. Los elementos de la Pila del Sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Propietario del Producto identifica los elementos de la Pila del Sprint que quiere ver completados y lo hace desde el conocimiento del equipo. Entonces, juntos determinan la cantidad de ese trabajo que puede comprometerse a completar durante el sprint. Durante el sprint, nadie puede cambiar la Pila del Sprint, lo que significa que los requisitos están congelados durante el sprint.

Un principio clave de SCRUM es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado *requirements churn*), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, SCRUM adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

El Propietario del Producto representa la voz del cliente. Se asegura que el equipo SCRUM trabaja de forma adecuada desde la perspectiva del negocio. Escribe historias de usuario, las prioriza y las coloca en la Pila del Producto.

El trabajo primario del Scrum Manager es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. No es el líder del equipo (porque ellos se autoorganizan), sino que actúa como una protección entre el equipo y cualquier influencia que lo distraiga. El Scrum Manager se asegura de que el proceso SCRUM se utilice como es debido. Scrum Manager es el que hace que las reglas se cumplan.

Equipo de desarrollo:

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etcétera).

Roles Auxiliares: Los roles auxiliares en los "equipos Scrum" son aquellos que no tienen un rol formal y no se involucran frecuentemente en el "proceso Scrum", sin embargo, deben ser tomados en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (*stakeholders*). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

Reuniones en Scrum: cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama *daily standup*. El SCRUM tiene unas guías específicas:
La reunión comienza puntualmente a su hora. La reunión tiene una duración fija de 15 minutos, de forma independiente al tamaño del equipo. Todos los asistentes deben mantenerse de pie (esto ayuda a mantener la reunión corta). La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días. Durante la reunión, cada miembro del equipo contesta a tres preguntas:

- ¿Qué has hecho desde ayer?
- ¿Qué es lo que estás planeando hacer hoy?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

Reunión de Revisión del Sprint (*Sprint Review Meeting*):

- Revisar el trabajo que fue completado y no completado.
- Presentar el trabajo completado a los interesados (alias «demo»).
- El trabajo incompleto no puede ser demostrado.

FERNANDO J. MARTINI

TECNOLOGIA DE LA INFORMACION

VOLUMEN 4

TELEINFORMATICA

TECNOLOGÍA DE LA INFORMACIÓN
VOLUMEN 4

TELEINFORMÁTICA

Versión: 1.01

Martini, Fernando J.
Tecnología de la Información : teleinformática . - 1a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2014.
v. 4, CD-ROM.
ISBN 978-987-33-4695-8
1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

Martini, Fernando J.
Tecnología de la Información. - 1a ed. - Ciudad Autónoma de Buenos Aires : el autor, 2014.
v.3, CD-ROM.
ISBN 978-987-33-4691-0
1. Informática. 2. Tecnología de la Información. I. Título
CDD 005.3

ÍNDICE

Capítulo III: Formas de transmisión, tipos de transmisión, tipos de enlaces, equipos de transmisión y software para la transmisión (protocolo)

1-La transmisión sincrónica y asincrónica
1.1-Sincrónica
1.2-Asincrónica
2-Tipos de transmisión
2.1-*Simplex*
2.2-*Duplex*
2.3-*Full Duplex*
3-Tipos de enlaces
3.1-Enlace punto a punto
3.2-Enlace punto a multipunto
3.3-Enlace multipunto a multipunto
4-Equipos de transmisión y software para la transmisión (protocolo)

Capítulo III: Redes de datos

1-Topología de redes
1.1-Topología en estrella
1.2-Topología de anillo
1.3-Topología de bus
1.4-Topología celular
2-Protocolo
2.1-Protocolos no orientados y orientados a la conexión
3-Red de comunicaciones
3.1-*Circuit Switching*
3.2-*Paquet* Switching o conmutación de paquetes
4-El modelo OSI
4.1-La capa física
4.3-Capa de red
4.4-Capa de transporte
4.5-Capa de sesión
4.6-Capa de presentación
4.7-Capa de aplicación
4.8-Diagrama de interacción entre las capas
5-Encapsulamiento
6-El modelo TCP/IP

7.7.9-IP sobre ATM

7.7.10-Mejoras de hardware para soluciones IP sobre ATM

Capítulo I
Conceptos elementales

1-Las comunicaciones

Tradicionalmente, la comunicación se ha definido como «el intercambio de sentimientos, opiniones o cualquier otro tipo de información mediante el habla, la escritura u otro tipo de señales». También, se la define como «el proceso mediante el cual se puede transmitir información de una entidad a otra». Los procesos de comunicación son interacciones mediadas por signos entre al menos dos agentes que comparten un mismo repertorio de signos y tienen unas reglas comunes.

Los dos conceptos básicos de las comunicaciones son la señal y el medio de transmisión:

La señal

Es la forma en que se representa la información. La señal tiene una semántica que establece las condiciones necesarias para que un signo pueda aplicarse a un objeto y las reglas que aseguran una significación lo más exacta posible.

Ej.: La escritura española, el ring de un celular, intervalos de humo, etc.

El medio de transmisión

Es el soporte por el cual se transmite la señal.

Ej.: El aire para ondas acústicas u ondas electromagnéticas, un cable para ondas eléctricas, un papel para la escritura española, etc.

1.1-La voz humana

La voz humana expresada en una determinada lengua es la señal más antigua. Es la forma de comunicación que más han utilizado los seres humanos. La semántica de la voz está dada por la variación de ondas generadas sobre un medio elástico (fundamentalmente el aire o un fluido). Estas ondas y sus variaciones deben ser posibles de ser percibidas por el oído y transmitidas al cerebro.

En la propagación en medios compresibles como el aire, implica que en algunas zonas las moléculas de aire, al vibrar, se juntan; y en otras zonas, se alejan; esta alteración de distancias entre las moléculas de aire es lo que produce el sonido. Lo mismo ocurre con las moléculas de agua en el caso de los sonidos transmitidos en el agua.

La frecuencia (Ver Figura 1) es el número de vibraciones u oscilaciones completas que se efectúan por segundo (hercios, Hz).

Los sonidos producidos son audibles por un ser humano promedio si la frecuencia de oscilación está comprendida entre 20 Hz y 20.000 Hz. Por encima de esta última frecuencia, se genera ultrasonido no audible por los seres humanos. Los sonidos con menos frecuencia se denominan graves y los de mayor frecuencia agudos. Mediante estas oscilaciones, se manifiesta el lenguaje.

La intensidad auditiva está dada por la amplitud (Ver Figura 1) de la onda (sonidos fuertes o débiles). La intensidad auditiva se mide en decibeles (dB). Los sonidos que perciben los seres humanos deben superar el umbral auditivo (0 dB) y no llegar al umbral de dolor (140 dB).

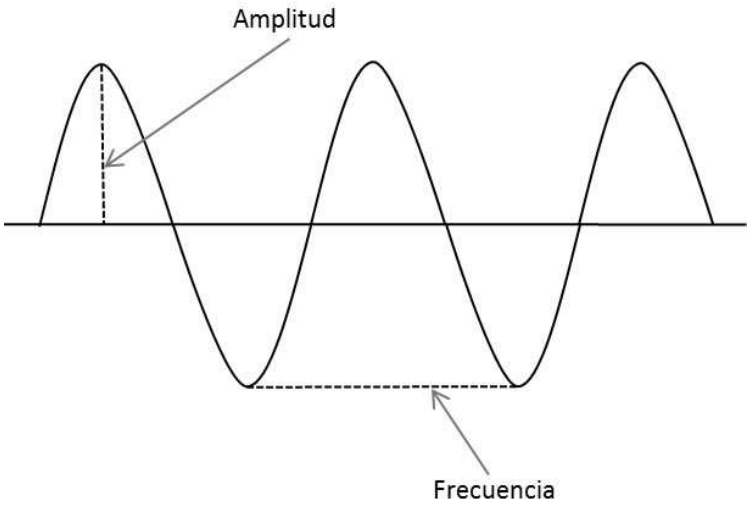


Figura 1

2-Telecomunicaciones

La palabra telecomunicaciones proviene del griego *tele* que significa «distancia», por lo tanto, el concepto se refiere a las comunicaciones a distancia. Además, este

mismo concepto involucra la utilización de la electrónica para trasladar las señales a distancia.

2.1-Las señales en las telecomunicaciones

Para abordar este tema con mayor capacidad de entendimiento, se aconseja ver el volumen 1 de esta colección.

Existen dos tipos básicos de señales que se utilizan en las telecomunicaciones: las señales analógicas y las señales digitales.

Señales analógicas

Una señal analógica puede verse como una forma de onda que toma un continuo de valores en cualquier momento dentro de un intervalo de tiempo. Por ejemplo, utilizando un micrófono, es posible pasar la onda acústica de la voz humana a una pequeña señal eléctrica, cuyo nivel de tensión siga una analógica con la variación sonora en volumen y frecuencia (Ver Figura 2). Esa onda eléctrica luego puede ser tomada por un altavoz y convertida nuevamente a una onda acústica.

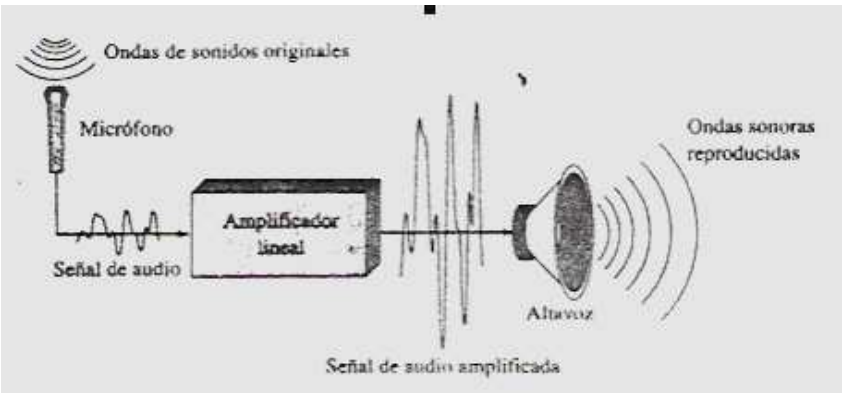


Figura 2

Señales digitales

La señal electrónica digital es muy simple, ya que su objetivo es transmitir dos tipos de estados, 0 o 1, bajo o alto o como quiera designarse a los estados binarios.

Por ejemplo: las señales eléctricas analógicas de la gráfica anterior pueden ser digitalizadas tomando muestras del valor de la señal y pasando cada uno de esos

valores a un valor binario; y luego, en lugar de trasmitir una señal analógica, se trasmite una señal digital (Ver Figuras 3 y 4). En la figura 3, se pude ver, en la parte superior, una onda analógica, y en la parte inferior, los puntos de muestreo que serán tomados de esa onda para luego ser digitalizados. En la figura 4, se puede ver la señal digital que será trasmitida, utilizando 8 bits para expresar cada punto del muestreo seleccionado.

Por ejemplo, una onda digital puede ser de entre 3 y 5 voltios para representar un 1, y de entre 0 y 1 voltio para representan un 0. La parte superior de la figura 4 muestra la representación binaria de un punto de la muestra. La imagen inferior muestra los valores, en voltios, reales de la onda que representa los ceros y unos.

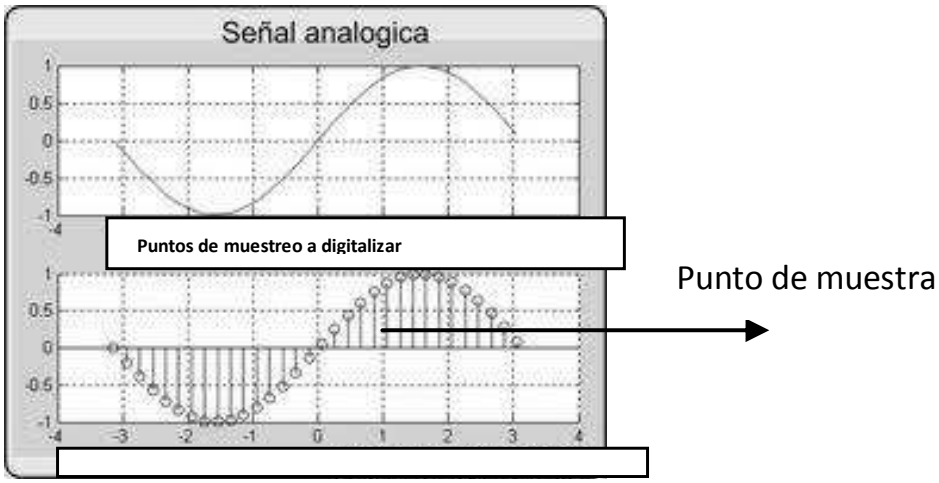


Figura 3

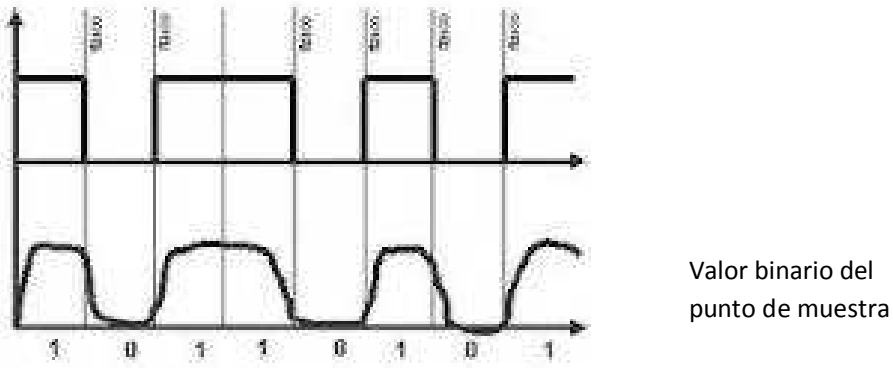


Figura 4

Una señal eléctrica puede enviar información analógica o digital, de igual forma lo puede hacer una onda electromagnética u otro tipo de señales. Lo que varía es la forma en que se utiliza la señal.

3-Teleinformática

La informática estudia métodos, procesos, técnicas, con el fin de almacenar, procesar y transmitir información en formato digital.

La teleinformática se pude definir como una disciplina que surge de la evolución y fusión de la telecomunicación y de la informática.

4-El sistema telefónico

La comunicación telefónica consiste en un equipo electrónico, el teléfono, que se utiliza para convertir la onda acústica en una onda eléctrica (ambas análogas) para que luego, en el teléfono del receptor, esa onda eléctrica vuelva a ser convertida a onda acústica (Ver Figura 5).

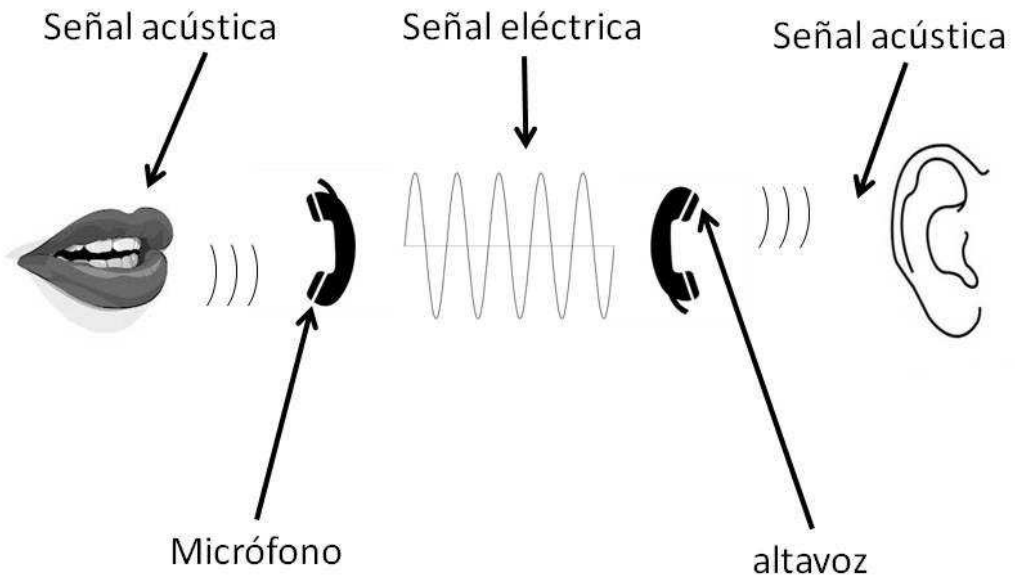


Figura 5

4.1-Líneas telefónicas

La primera tecnología utilizada en la teleinformática fue la utilización de las líneas telefónicas para conectar computadoras.

El esquema de esta modalidad es la siguiente (Ver Figura 6):

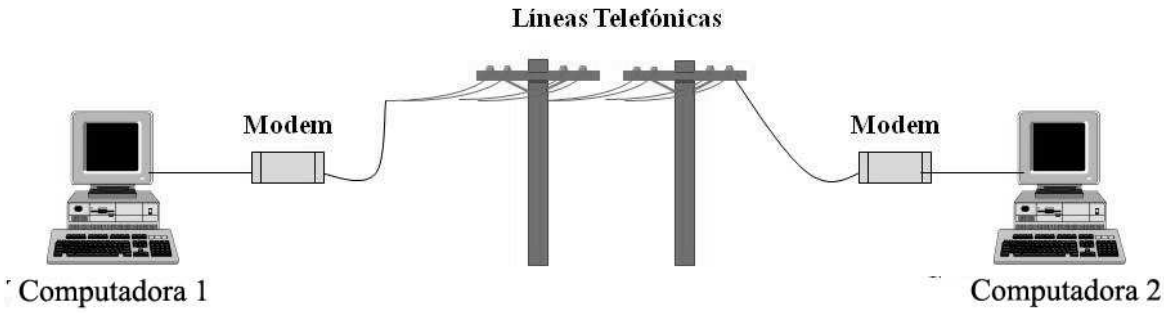


Figura 6

Este esquema funciona de la siguiente forma:

- La computadora 1 emite una señal digital (son señales simples de decodificar en ceros y unos) (Ver Figura 4).

- La señal digital es modulada por el módem en una señal analógica (el motivo por el cual la señal es pasada de digital a analógica es un problema de ancho de banda que se entenderá más adelante). Las señales analógicas son más complejas en el sentido de poder codificar o decodificar los ceros y los unos.

- El módem demodula la señal y la transforma en digital.

- La señal digital es capturada por la computadora 2.

Este esquema funciona de igual forma cuando la computadora 2 envía una señal a la computadora 1.

4.2-El módem

El módem (modulador-demodulador) es un equipo electrónico cuya finalidad es pasar la señal digital a señal analógica (modular) y viceversa (demodular). El motivo por el cual es necesario modular la señal digital es que las señales analógicas sufren una menor atenuación y distorsión que las señales digitales, y de esta forma pueden recorrer mayores distancias antes de que la atenuación y la distorsión ocasionen la imposibilidad de recuperar la señal. Además, las líneas telefónicas fueron diseñadas para transmitir señales analógicas cuyo intervalo de frecuencias varía entre 300 y 3.300 Hz.

El módem emite una señal analógica que se denomina portadora, esta es constante en amplitud, frecuencia y fase (más adelante se explicará detalladamente el concepto de fase). Esta señal portadora es modulada para codificar los ceros y los unos variando la amplitud o la frecuencia, o la fase, o una combinación de ambas. Dentro de cada una de estas formas básicas, existen diferentes técnicas que no se explicarán, porque exceden el alcance de este libro, solamente se verá una explicación muy simple de los métodos ASK, FSK y PSK.

4.2.1-Por amplitud

Una de las técnicas utilizadas en la modulación por amplitud se llama *Amplitude Shift Keying* (ASK). Los unos y ceros están representados por diferencias en la amplitud de la señal (Ver Figura 7). Ciclos de igual amplitud (voltaje) determinan un 0 o un 1. La velocidad de transmisión está limitada por las características físicas del medio de transmisión.

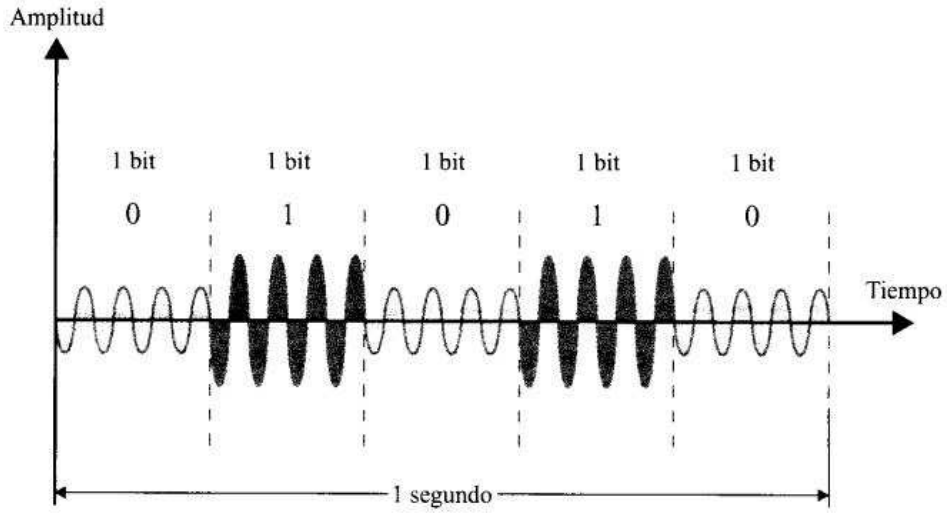


Figura 7

4.2.2-Por frecuencia

Una de las técnicas utilizadas en la modulación por frecuencia se llama *Frequency Shift Keying* (FSK). Los unos y ceros están representados por diferencias en la frecuencia de la señal (Ver Figura 8). Los ciclos de igual frecuencia (ciclos por segundo) determinan un 0 o un 1.

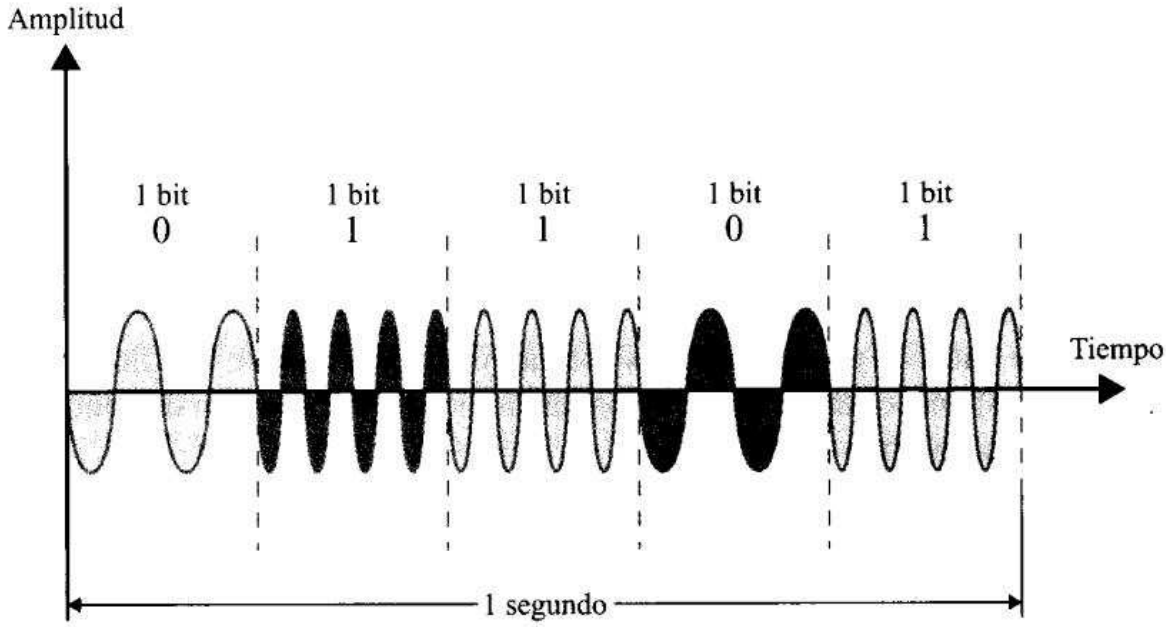


Figura 8

4.2.3-Por fase

En la figura 9, se puede ver un alternador precario, es un material conductor de forma rectangular que está conectado a unas escobillas por su eje de rotación. Al rotar la espira entre los imanes, se genera electricidad inducida.

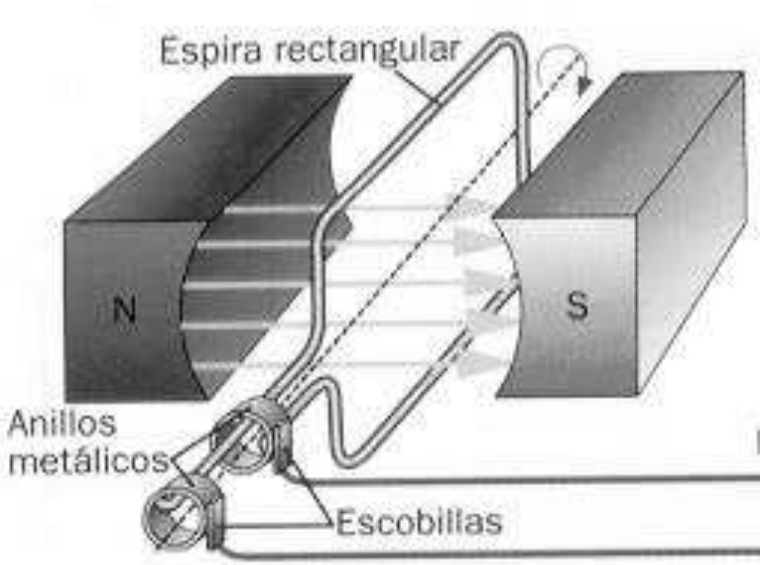


Figura 9

Una rotación completa (360°) de la espira genera una tensión senoidal como la que se muestra en la figura 10.

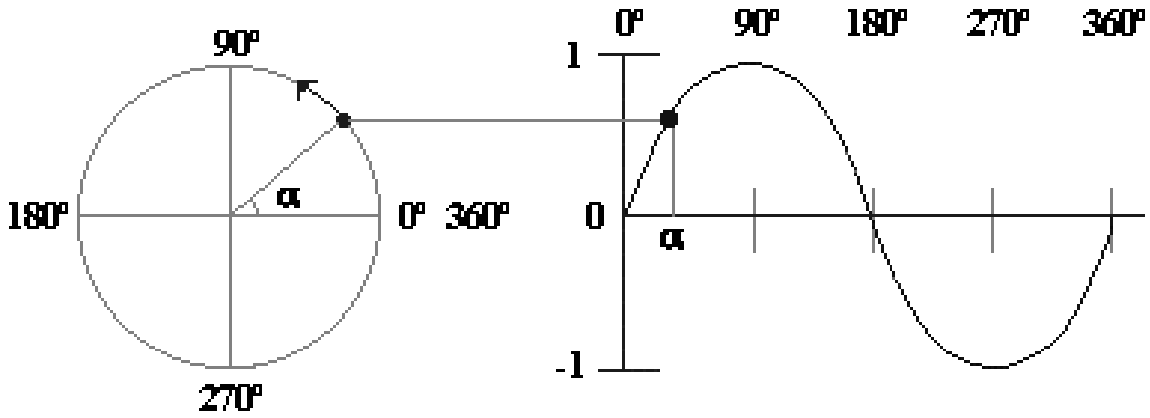


Figura 10

Mientras que la espira rote en una velocidad constante, la frecuencia y la fase se mantendrán constantes, como en la figura 10.

Se dice que dos señales medidas en el mismo momento están en fase si las dos tienen el mismo grado. Por el contrario, si las dos señales tienen distinto grado**,** se dice que están desfasadas (Ver Figura 11).

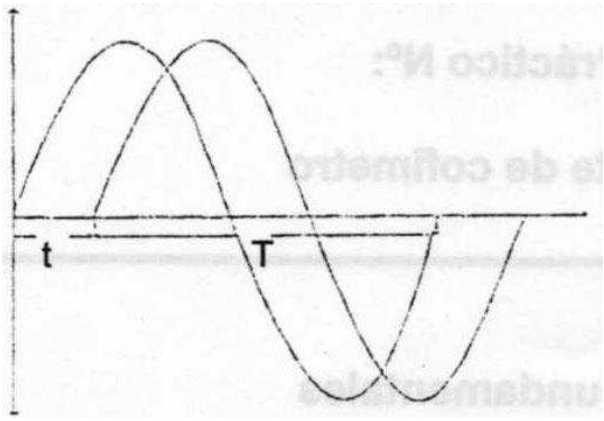


Figura 11

En otros términos, el ángulo de desfasaje, entre dos señales eléctricas senoidales distintas de igual frecuencia, es la diferencia de grados medidos en el mismo punto de tiempo.

Ahora es posible entrar en el concepto de codificación por fase. En la técnica *Phase Shift Keying* (PSK), los ceros y los unos se codifican por un desplazamiento de la fase en la modulación de la señal eléctrica. Por ejemplo, una fase puede comenzar en 0° para codificar un 0 binario y cambiar a 180° para identificar el cambio de estado binario y representar un 1 (Ver Figura 12).

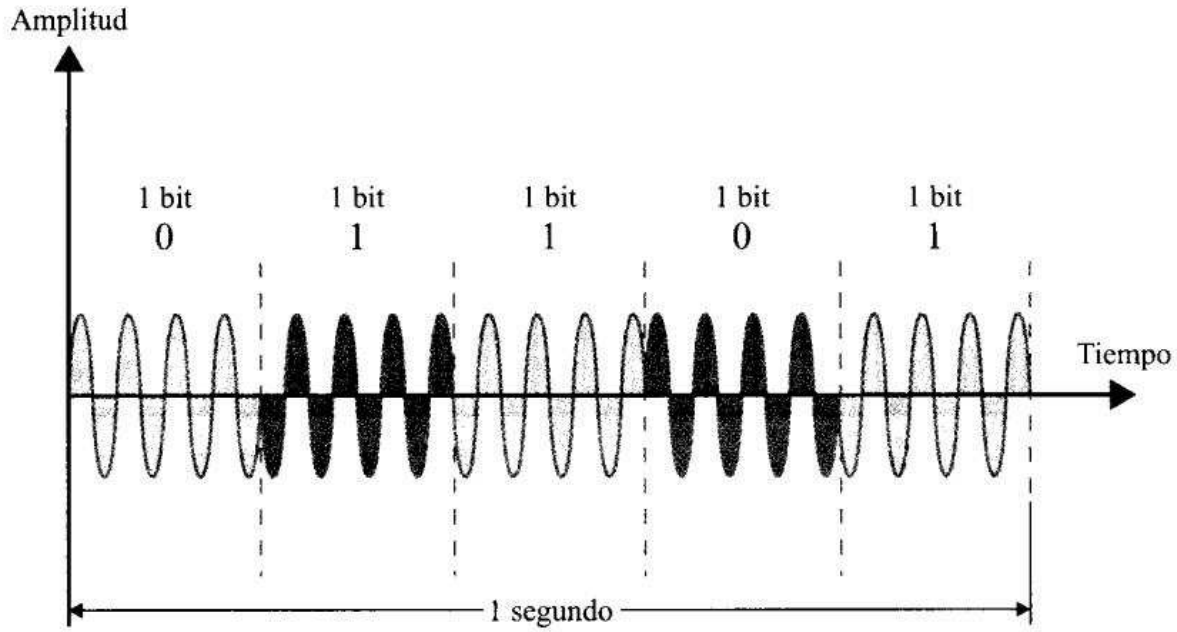


Figura 12

5-Componentes de un sistema de telecomunicaciones

Los componentes de un sistema de telecomunicaciones son:

- Computadoras
- Medios de transmisión
- Equipos y circuitos específicos para comunicaciones
- Métodos de envío de señal (protocolos)

Estos elementos serán desarrollados a lo largo del libro.

6-El ancho de banda de un medio de transmisión

El ancho de banda de un medio de transmisión se puede definir como el espacio entre la frecuencia superior e inferior que se puede transmitir sin distorsión por un medio de transmisión.

7-Velocidad de transferencia

La velocidad de transmisión, también denominada tasa de bits, se refiere a la cantidad de bits que pueden transmitirse entre equipos digitales en un período de

tiempo. Normalmente se utiliza el segundo como espacio de tiempo y se denomina bits por segundo (bps, la b debe ser siempre minúscula para no confundirla con los Bytes).

La escala de medición es la siguiente:

- Kbps o mil bits por segundo (Kilobit por segundo)
- Mbps o un millón de bits por segundo (Megabit por segundo)
- Gbps o mil millones de bits por segundo (Gigabit por segundo)
- Etcétera

Existen muchos factores que afectan la velocidad de transmisión, entre ellos se pueden destacar (ahora se enuncian, pero en el transcurso de este volumen se explicará cada uno de ellos):

- El ancho de banda
- La distancia entre los equipos
- Los equipos de comunicación
- La topología de la red
- La cantidad de usuarios de la red

Los proveedores de servicios de comunicaciones muchas veces utilizan el término ancho de banda como sinónimo de velocidad de transferencia.

Capítulo II
Medios de transmisión

Un medio de transmisión es un canal que posibilita la transmisión de información. Se los clasifica en dos grandes grupos: guiados y no guiados.

1-Medios de transmisión guiados

Se denominan medios de transmisión guiados a aquellos en los que la información es guiada por un medio físico hacia un punto determinado y son los siguientes:

- Pares trenzados
- Cable coaxial
- Fibra óptica

1.1-Pares trenzados

El cable de pares trenzados se produce con dos hilos de un material conductor, generalmente cobre, cada uno recubierto por un material aislante. Los hilos se trenzan con el objetivo de evitar interferencias. Los cables de par trenzado no son caros y son fáciles de conectar. Tienen la desventaja de tener que usarse a distancias limitadas, ya que la señal se va atenuando muy rápidamente. Por este motivo, a corta distancia, se debe emplear repetidores que restauren la señal.

Existen muchas variantes de pares trenzados, las más destacadas son: los *Unshielded Twisted Pair* (UTP), los *Foiled Twisted Pair* (FTP) y los *Shielded Twisted Pair* (STP).

1.1.1-*Unshielded Twisted Pair* (UTP)

El UTP es un cable que no tiene revestimiento o blindaje entre la cubierta exterior y los cables (Ver Figura 13).

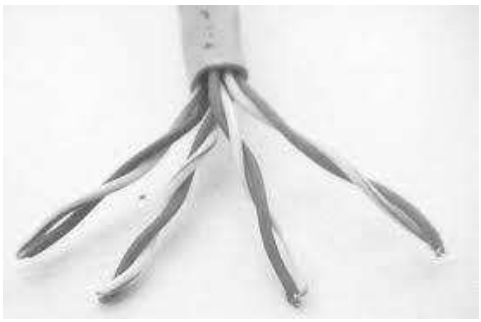


Figura 13

Se comercializan cinco categorías de cables UTP (Ver Figura 14), cada uno de ellos con un uso específico (básicamente según el protocolo) que podrá comprenderse más adelante.

Tipo	Uso
Categoría 1	Voz solamente (cable telefónico)
Categoría 2	Datos hasta 4 Mbps (LocalTalk)
Categoría 3	Datos hasta 10 Mbps (Ethernet)
Categoría 4	Datos hasta 20 Mbps (16 Mbps Token Ring)
Categoría 5	Datos hasta 100 Mbps (Fast Ethernet)
Categoría 5e	Datos hasta 1000 Mbps (Fast Ethernet)
Categoría 6	Datos hasta 10 Gbps (Fast Ethernet) en 250 MHz
Categoría 6a	Datos hasta 10 Gbps (Fast Ethernet) en 500 MHz
Categoría 7	Datos hasta 10 Gbps (Fast Ethernet) en 1000 MHz

Figura 14

1.1.2-*Foiled Twisted Pair* (FTP)

El FTP es un cable de par trenzado similar al UTP, pero, a diferencia de este, los pares de cables trenzados son recubiertos por una malla externa de aluminio o de cobre

trenzado o un papel metálico alrededor del conjunto de pares. Esta malla metálica o papel metálico tiene como objetivo disminuir los ruidos eléctricos (Ver Figura 15).

Figura 15

1.1.3-*Shielded Twisted Pair* (STP)

El STP es un cable de par trenzado similar al UTP, pero, a diferencia de este, cada par tiene una cubierta protectora (papel metálico), además de tener una malla externa de aluminio o de cobre trenzado alrededor del conjunto de pares. Estas dos cubiertas son diseñadas para reducir la absorción del ruido eléctrico. Este cable es más costoso y difícil de manipular que el cable sin blindaje. Es muy apropiado para ambientes donde puede haber altísimo ruido eléctrico, como entre máquinas industriales (Ver Figura 16).

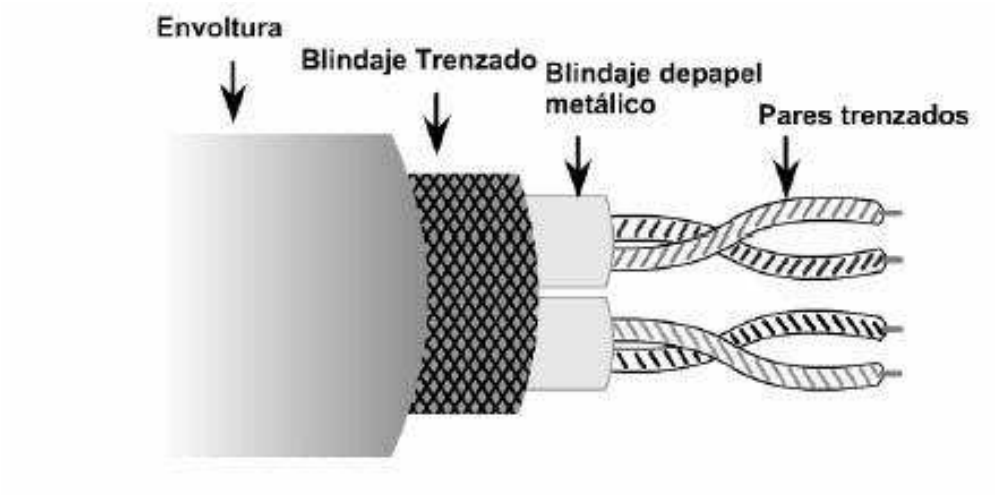


Figura 16

1.2-Cable coaxial

Es un cable utilizado para transportar señales eléctricas de alta frecuencia. Este posee dos conductores concéntricos: uno central, llamado vivo, encargado de llevar la información; y uno exterior, de aspecto tubular, llamado malla o blindaje, que sirve como referencia de tierra y retorno de las corrientes. Entre ambos se encuentra una capa aislante llamada dieléctrico, de cuyas características dependerá principalmente la calidad del cable. Todo el conjunto suele estar protegido por una cubierta aislante (Ver Figura 17).

Figura 17

Existen múltiples tipos de cable coaxial, cada uno con un diámetro e impedancia diferentes. El cable coaxial no es habitualmente afectado por interferencias externas y es capaz de lograr altas velocidades de transmisión en largas distancias. Por esa razón, se utiliza en redes de comunicación de banda ancha. Con este cable se puede llegar a transmitir datos de hasta 200 Mbps.

1.3-Fibra óptica

Consiste en un cable conductor de sílice (SiO2 –dióxido de silicio– es uno de los compuestos de la arena, por lo tanto, muy abundante) o plástico (solo en cortas distancias) que transmite impulsos luminosos normalmente emitidos por un láser o un LED. En este cable, el rayo de luz es direccionado dentro de él por un antiguo sistema denominado «refracción». Por ejemplo, en las viejas pirámides de Egipto, la luz se transmitía a su interior utilizando el mecanismo de refracción lumínica que se efectuaba sobre una serie de espejos colocados estratégicamente para direccionar la luz.

La señal eléctrica recibida es transformada en luminosa a través de un dispositivo eléctrico-óptico, como un láser o un LED. El grosor de una fibra óptica es de aproximadamente unos 125 µm (micrómetros). Para hacer una comparación, un cabello humano tiene unos 70 µm de diámetro.

Un cable de fibra óptica está compuesto por varias fibras ópticas. Los cables de fibras ópticas comparten su espacio con hilos de aramida (fibra sintética, robusta y resistente al calor) que le confieren la necesaria resistencia a la tracción. Un cable con 8 fibras ópticas tiene un tamaño bastante más pequeño que los utilizados habitualmente (Ver Figura 18)

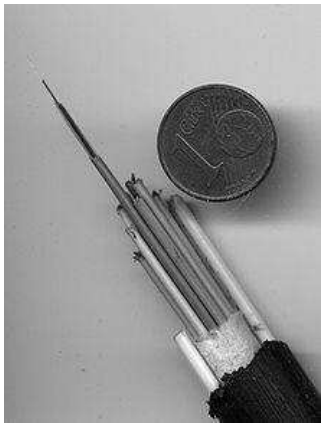


Figura 18

La fibra óptica está formada por tres capas (Ver Figura 19):

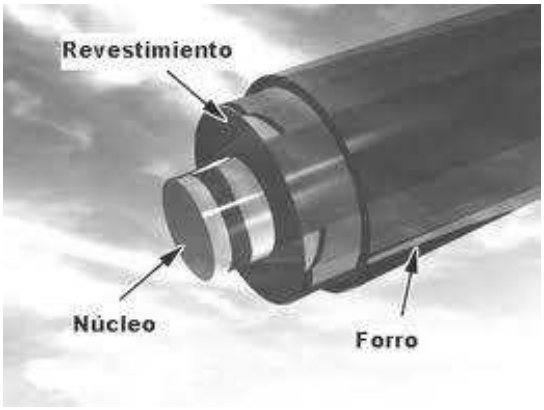


Figura 19

- El núcleo: por el cual se propagan las ondas de luz.

- El revestimiento del núcleo: generalmente del mismo material que el núcleo, pero con algún aditivo agregado que le cambia el índice de refracción y de esta forma confina la onda de luz al conducto central o núcleo (Ver Figura 20).

- El forro: por lo general está fabricado en plástico y asegura la protección mecánica de la fibra.

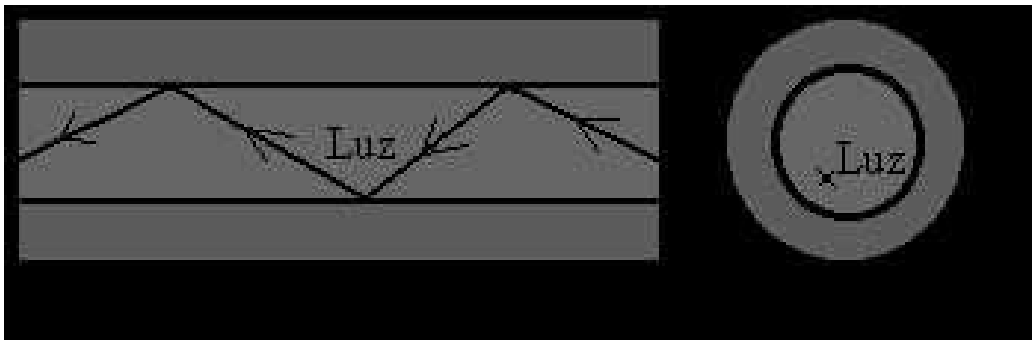


Figura 20

Las diferentes trayectorias que puede seguir un haz de luz en el interior de una fibra se denominan modos de propagación. Y según el modo de propagación, existen dos tipos de fibra óptica: monomodo y multimodo.

1.3.1-Monomodo

Una fibra monomodo es una fibra óptica en la que solo se propaga un modo de luz (paralela al eje de la fibra). Esto se logra reduciendo el diámetro del núcleo de la fibra hasta un tamaño tal (8,3 a 10 micrones) que solo permite un modo de propagación. Las fibras monomodo permiten alcanzar grandes distancias (hasta 400 km máximo, mediante un láser de alta intensidad) y transmitir elevadas tasas de información (Ver Figura 21).

Figura 21

En monomodo supera la velocidad de transmisión de 10 Gbit/s.

1.3.2-Multimodo

Una fibra multimodo es aquella en la que los haces de luz pueden circular por más de un modo o camino. Una fibra multimodo puede tener más de mil modos de propagación de luz. Las fibras multimodo se usan comúnmente en aplicaciones de corta distancia, menores a 2 km, son simples de diseñar y económicas.

Existen dos variantes:

- Índice escalonado: en este tipo de fibra, el núcleo tiene un índice de refracción constante en toda la sección cilíndrica y tiene alta dispersión modal (Ver Figura 22).

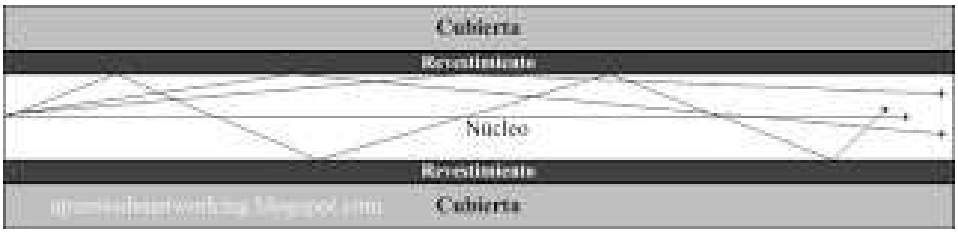


Figura 22

- Índice gradual: mientras que en este tipo, el índice de refracción no es constante, tiene menor dispersión modal y el núcleo se constituye de distintos materiales (Ver Figura 23).

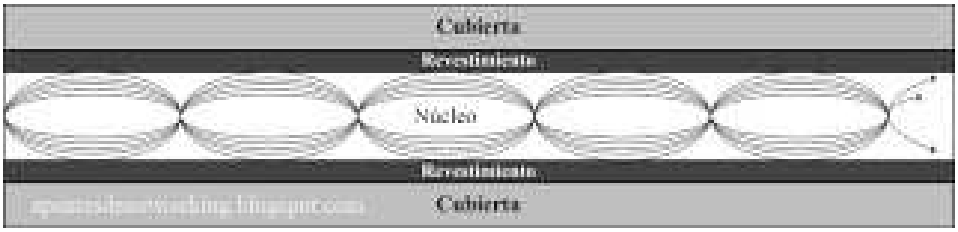


Figura 23

En multimodo las velocidades dependen de sobre qué tecnología se esté trabajando. En los estándares OM1 y OM2 (multimodo que usa LED), en protocolo Ethernet, llegan a 1 Gbit/s. En el estándar OM3 (multimodo que usa láser), en protocolo Ethernet, llegan a 10 Gbit/s.

2-Medios de transmisión no guiados

En este tipo de medios de transmisión, las ondas no son dirigidas por un cable, sino que son lanzadas al aire con la ayuda de antenas para transmisión y tomadas del aire por antenas receptoras. El portador de la información entre la antena emisora y la receptora son ondas electromagnéticas. Esta transmisión es mucho más susceptible de recibir interferencias y ruidos que utilizando medios guiados.

Una onda electromagnética es un tipo de radiación en forma de onda que se caracteriza por poseer dos campos: un campo eléctrico y otro campo magnético, oscilando perpendicularmente entre sí (Ver Figura 24).

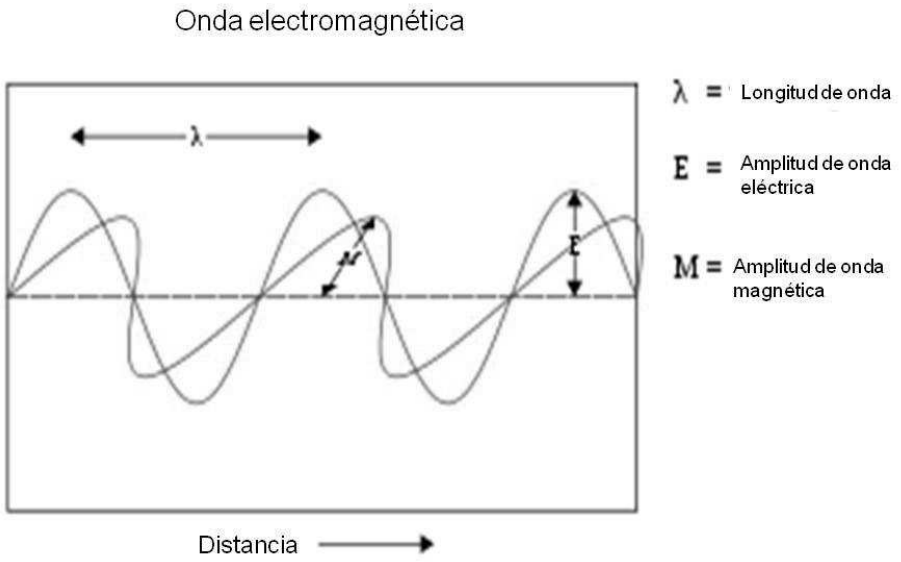


Figura 24

Siguen a continuación una serie de conceptos que es necesario repasar para entender mejor el comportamiento de estas ondas y sus aplicaciones.

Ciclo: se denomina ciclo a cada patrón repetitivo de una onda.

Período: es el tiempo que tarda la onda en completar un ciclo.

Frecuencia: es el número de ciclos que completa la onda en un intervalo de tiempo. Si dicho intervalo es de un segundo, la unidad de frecuencia es el Hercio (Hz).

Amplitud: es la medida de la magnitud de la máxima perturbación del medio producida por la onda.

Longitud: la longitud de una onda está determinada por la distancia entre los puntos inicial y final de un ciclo (por ejemplo, entre un valle de la onda y el siguiente). Habitualmente se denota con la letra griega lambda (Λ).

Velocidad: las ondas se desplazan a una velocidad que depende de la naturaleza de la onda y del medio por el cual se mueven. En el caso de la luz, por ejemplo, la velocidad en el vacío se denota «c» y vale 299.792.458 m/s (aproximadamente 3.10^8 m/s).

Polarización: la polarización representa la orientación de cómo la onda oscila, y en el caso particular de las ondas electromagnéticas, la orientación en la oscilación del campo eléctrico. A menudo esta orientación es una línea y por ello se habla típicamente de ondas con polarización vertical u horizontal, es decir, cuando el campo eléctrico oscila en un plano con esas direcciones.

Se denomina espectro electromagnético a todo el rango posible de radiación electromagnética. Esto incluye las ondas de radio, los rayos infrarrojos, la luz, los rayos ultravioletas, los rayos X, los rayos gamma, etcétera.

En función de lo anterior, el espectro radioeléctrico o de Radio Frecuencia (RF) se refiere a la porción del espectro electromagnético en el cual las ondas electromagnéticas pueden generarse alimentando a una antena con corriente alterna.

La tabla a continuación presenta las bandas de RF más importantes:

Frecuencia	**Abreviatura**	**Frecuencia y longitud de onda (aire)**	**Ejemplos de uso**
Tremendamente baja	TLF	< 3 Hz > 100,000 km	Ruido natural o provocado por el hombre.
Extremadamente baja	ELF	3–30 Hz 100,000 km – 10,000 km	Comunicación con submarinos.

Súper baja	SLF	30–300 Hz 10,000 km – 1000 km	Comunicación con submarinos.
Ultra baja	ULF	300–3000 Hz 1000 km – 100 km	Comunicación con submarinos, comunicaciones en minas a través de la tierra.
Muy baja frecuencia	VLF	3–30 kHz 100 km – 10 km	Radioayuda, señales de tiempo, comunicación submarina, pulsómetros inalámbricos, geofísica.
Baja frecuencia	LF	30–300 kHz 10 km – 1 km	Radioayuda, señales de tiempo, radiodifusión en AM (onda larga) (Europa y partes de Asia), RFID, radioafición.
Frecuencia media	MF	300–3000 kHz 1 km – 100 m	Radiodifusión en AM (onda media), radioafición, balizamiento de aludes.
Alta frecuencia	HF	3–30 MHz 100 m – 10 m	Radioafición en Onda corta, banda ciudadana y radioafición, comunicaciones de aviación sobre el horizonte, RFID, radar, comunicaciones ALE, comunicación cuasivertical (NVIS), telefonía móvil y marina.
Muy alta frecuencia	VHF	30–300 MHz 10 m – 1 m	FM, televisión, comunicaciones con aviones a la vista entre tierra-avión y avión-avión, telefonía móvil marítima y terrestre, radioaficionados, radio meteorológica.
Ultra alta frecuencia	UHF	300–3000 MHz 1 m – 100 mm	Televisión, hornos microondas, comunicaciones por microondas, radioastronomía, telefonía móvil, redes inalámbricas, bluetooth, ZigBee, GPS, comunicaciones uno a uno como FRS y GMRS, radioafición.
Súper alta frecuencia	SHF	3–30 GHz 100 mm – 10 mm	Radioastronomía, comunicaciones por microondas, redes inalámbricas, radares modernos, comunicaciones por satélite, televisión por satélite, DBS, radioafición.
Frecuencia extremadamente alta	EHF	30–300 GHz 10 mm – 1 mm	Radioastronomía, transmisión por microondas de alta frecuencia, teledetección, radioafición, armas de microondas, escáner de ondas milimétricas.
Terahercios o Frecuencia tremendamente alta	THz o THF	300–3,000 GHz 1 mm – 100 m	Radiografía de terahercios –un posible sustituto para los rayos X en algunas aplicaciones médicas–, dinámica molecular ultrarrápida, física de la materia condensada, espectroscopía mediante terahercios, comunicaciones/computación mediante terahercios.

2.1-Microondas

Se describe como microondas a aquellas ondas electromagnéticas cuyas frecuencias van desde los 500 MHz hasta los 300 GHz o aún más. Por consiguiente, las señales de microondas, a causa de sus altas frecuencias, tienen longitudes de onda relativamente pequeñas, de ahí el nombre de «micro» ondas. La gran mayoría de los

sistemas actuales de radio de microondas es de modulación de frecuencia, que es de naturaleza analógica. Sin embargo, se han elaborado nuevos sistemas que usan modulación por conmutación de fase o por amplitud en cuadratura, que son formas básicamente de modulación digital. De la tecnología que se utilice depende en gran medida los costos de los equipamientos y el servicio en general.

La antena utilizada generalmente en las microondas es la de tipo parabólico. El tamaño típico es de un diámetro de unos 3 m. La antena es fijada rígidamente y transmite un haz estrecho que debe estar perfectamente enfocado hacia la antena receptora.

Por ejemplo, dos antenas de microondas situadas a una altura de 100 m pueden separarse una distancia total de 82 km, ya que a más distancia la antena emisora y receptora dejarían de verse por la circunferencia terrestre (Ver Figura 25). Cualquier obstáculo entre las antenas impediría una transmisión confiable. Para superar la distancia de 82 km o los escoyos en el camino, se utilizan antenas retransmisoras.

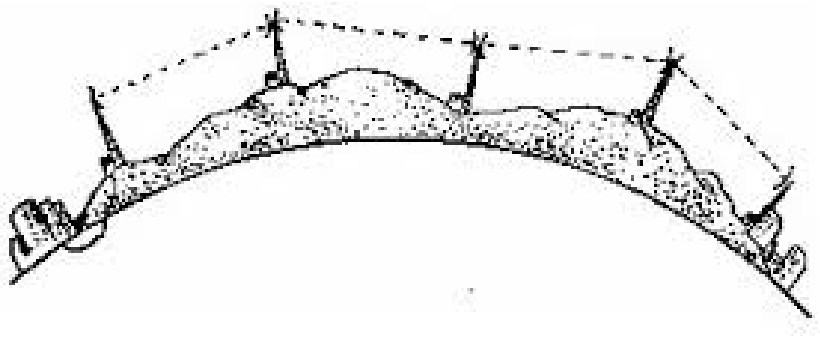


Figura 25

La transmisión por microondas puede llegar velocidades de 100 Mbps.

2.2-Vía satélite

Desde los inicios de la era espacial, con la puesta en órbita del primer satélite artificial, en 1957, los avances han sido constantes. Los satélites de comunicaciones tienen dos características muy importantes que los destaca sobre los otros medios de comunicación: una es su considerable ancho de banda y la otra, la posibilidad de lograr una cobertura global.

Todos los satélites de comunicaciones se sitúan en una órbita geoestacionaria, de modo que el satélite aparece como un punto fijo en el firmamento. Esta función es

lograda porque se colocan a una altura de 35.786 km donde la fuerza centrífuga de rotación (a la misma velocidad que gira la Tierra en el paralelo mayor, el Ecuador) se iguala con la fuerza de atracción de la gravedad. Esta particularidad del satélite geoestacionario permite que las antenas terrestres puedan estar fijas para direccionar las señales al satélite (Ver Figura 26).

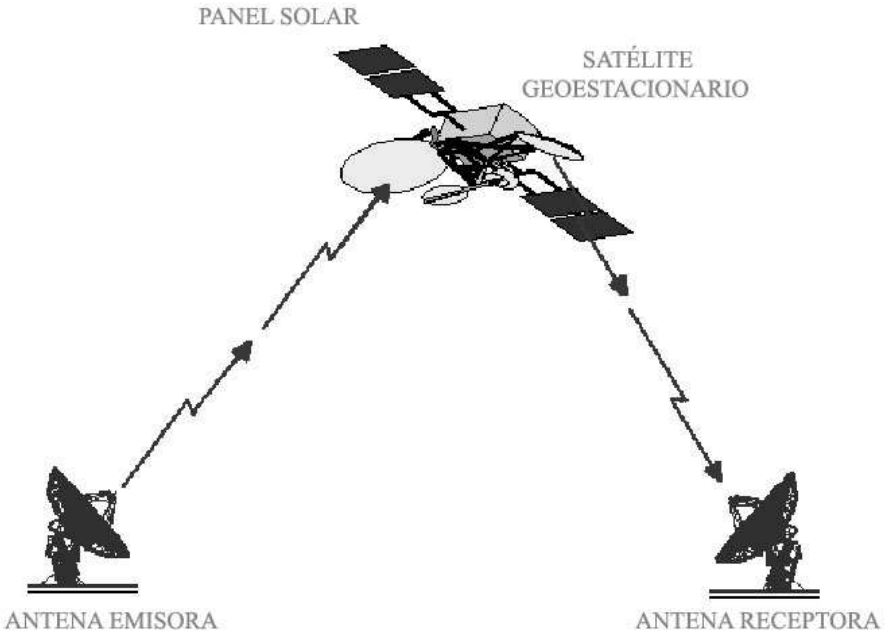


Figura 26

La transmisión se efectúa, al igual que la comunicación por microonda, mediante la utilización de señales electromagnéticas. Se establece una banda de subida y otra de bajada a los efectos de evitar la interferencia entre ellas (esto es igual que en las transmisiones terrestres).

La transmisión vía satélite puede llegar velocidades de 100 Mbps.

2.3-Telefonía móvil

El teléfono móvil es un dispositivo inalámbrico electrónico para acceder y utilizar los servicios de la red de telefonía celular o móvil. Se denomina celular debido a que el servicio funciona mediante una red de celdas, en la que cada antena repetidora de señal es una célula. Según las bandas o frecuencias en las que opera el móvil, podrá funcionar en una parte u otra del mundo.

En su operación, el teléfono móvil establece comunicación con una estación base y, a medida que se traslada, los sistemas computacionales que administran la red van

transmitiendo la llamada a la siguiente estación base de forma transparente para el usuario. Es por eso que se dice que las estaciones base forman una red de celdas que sirven cada estación base a los equipos móviles que se encuentran en su celda satélite (Ver Figura 27).

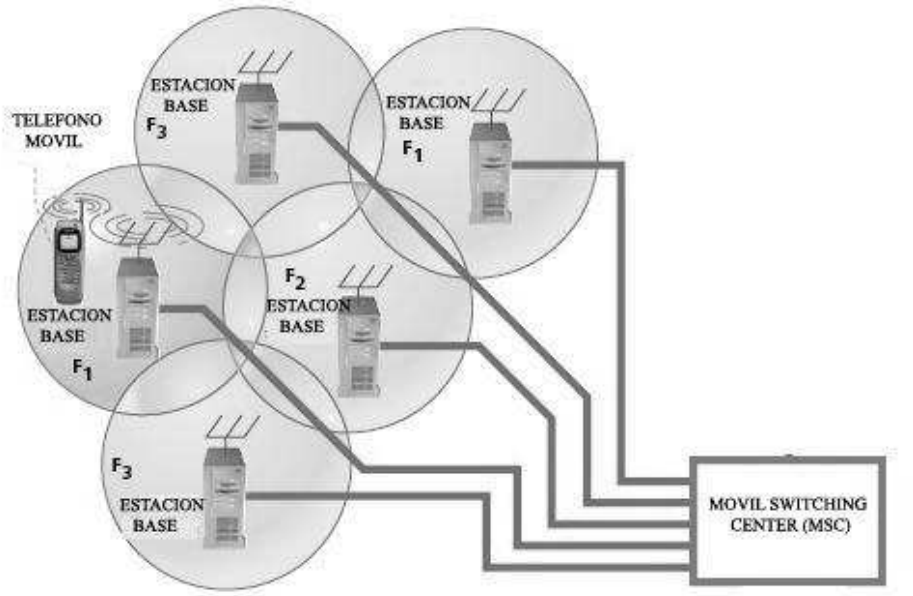


Figura 27

Capítulo III

Formas de transmisión, tipos de transmisión, tipos de enlaces, equipos de transmisión y software para la transmisión (protocolo)

Las formas y tipos de transmisión están en relación con todo el sistema de transmisión. No es solo el canal o el hardware utilizado, sino que es en función de:

- Computadoras
- Medios de transmisión
- Equipos y circuitos específicos para comunicaciones
- Métodos de envío de señal (protocolos)

1-La transmisión sincrónica y asincrónica

1.1-Sincrónica

La transmisión se basa en un mecanismo de relojería. Aquí es necesario que el transmisor y el receptor utilicen la misma frecuencia para identificar los caracteres. La técnica consiste en el envío de una trama de datos (conjunto de caracteres) que configura un bloque de información comenzando con un conjunto de bits de sincronismo (SYN) y terminando con otro conjunto de bits de final de bloque (ETB). En este caso, los bits de sincronismo tienen la función de sincronizar los relojes existentes tanto en el emisor como en el receptor, de tal forma que estos controlan la duración de cada bit y carácter. Por lo tanto, el ritmo de llegada de la información al destino tiene que coincidir con el ritmo de salida de la información de la fuente. Esto implica la necesidad de un sistema de transmisión que permita una velocidad considerablemente amplia como para mantener una velocidad de transmisión constante a través de todo el sistema.

Ventajas:

- Posee un alto rendimiento en la transmisión.
- Son aptos para transmisiones de altas velocidades.
- El flujo de datos es más regular que en el asincrónico.

1.2-Asincrónica

Es también conocida como start/stop. Requiere de una señal que identifique el inicio del carácter o bloque de caracteres y se la denomina bit de arranque. También se requiere de otra señal denominada señal de parada que indica la finalización del carácter o bloque de caracteres.

Una transmisión es asíncrona cuando no hay ninguna relación temporal entre la estación que transmite y la que recibe. Es decir que el ritmo de llegada de la información al destino no tiene por qué coincidir con el ritmo de salida de la información por la fuente. En estas situaciones, no es necesario garantizar un flujo de información a velocidad constante. En este tipo de transmisión, el receptor no sabe con precisión cuándo recibirá un mensaje.

Ventajas:

- Es un procedimiento que permite el uso de equipamiento más económico y de tecnología menos sofisticada.
- Se adecua más fácilmente en aplicaciones, cuyo flujo transmitido es más irregular.
- Son especialmente aptos cuando no se necesita lograr alta velocidad.

2-Tipos de transmisión

2.1-*Simplex*

Este tipo de transmisión permite que la información discurra en un solo sentido y de forma permanente, la comunicación es unidireccional y se emplean usualmente en redes de radiodifusión, como la señal de TV (Ver Figura 28).

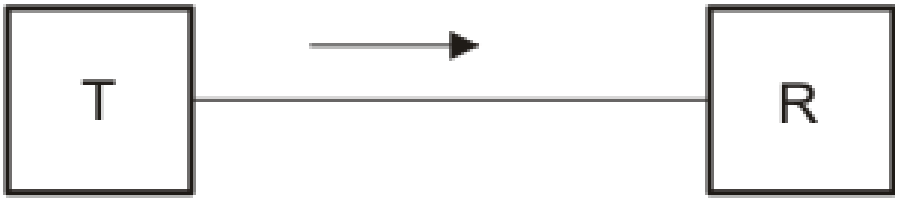


Figura 28

2.2-*Duplex*

En este tipo, la transmisión fluye en los dos sentidos, pero no simultáneamente. Una estación debe enviar y la otra recibir o viceversa. Este método también se denomina de dos sentidos alternos. Por ejemplo, en el *walkie-talkie* o, muy habitualmente, en la interacción entre terminales y un computador central (Ver Figura 29).

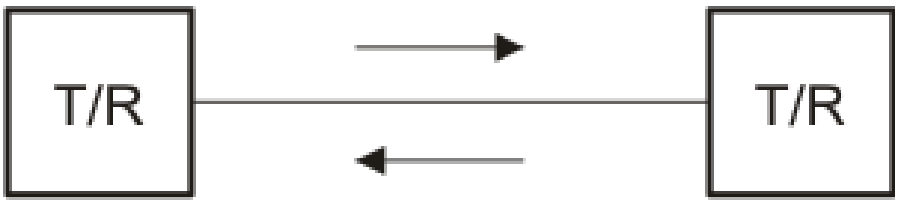


Figura 29

2.3-*Full Duplex*

Es muy similar al *duplex*, con la diferencia de que los datos pueden desplazarse en ambos sentidos simultáneamente. Para lograr dicho fin, los transmisores de ambos lados de la línea poseen diferentes frecuencias de transmisión o dos caminos de comunicación separados, mientras que la comunicación *duplex* necesita uno solo. Para intercambiar datos entre computadores, este tipo de comunicaciones es más eficiente que la transmisión *duplex*.

3-Tipos de enlaces

Existen tres tipos básicos para unir o enlazar nodos o equipos: punto a punto, punto a multipunto y multipunto a multipunto.

3.1-Enlace punto a punto

Los enlaces punto a punto son aquellos que responden a un tipo en que el canal de datos se usa para comunicar únicamente dos nodos. Las redes punto a punto son relativamente fáciles de instalar y de operar (Ver Figura 30).

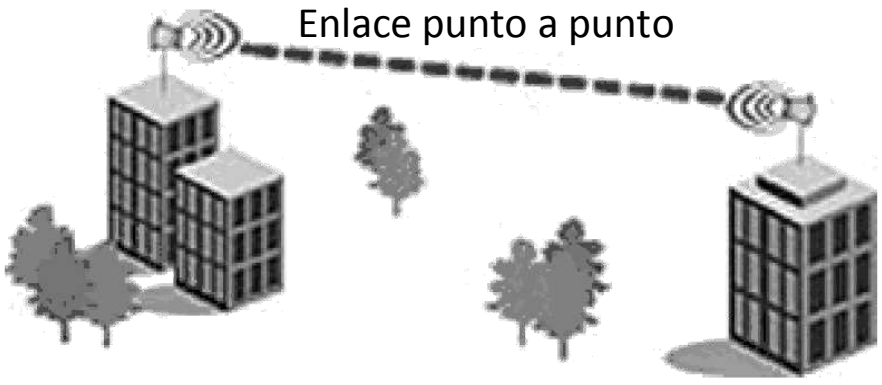


Figura 30

3.2-Enlace punto a multipunto

En un enlace punto a multipunto, existe un nodo (nodo 1) que se comunica directamente con un único otro nodo (nodo 2) y este a su vez se comunica con otro u otros nodos. En esta topología, el nodo 1, para enlazarse con cualquier otro nodo de la red, siempre deberá pasar por el nodo 2 (Ver Figura 31).

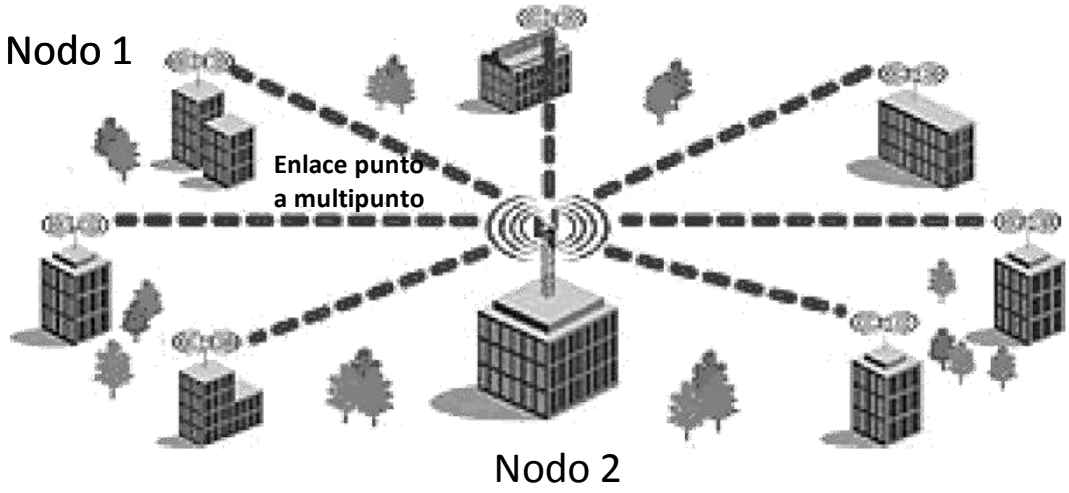


Figura 31

3.3-Enlace multipunto a multipunto

Este tipo de enlace consiste en que ambos nodos de la conexión se comunican con otro u otros nodos de la malla o red (Ver Figura 32).

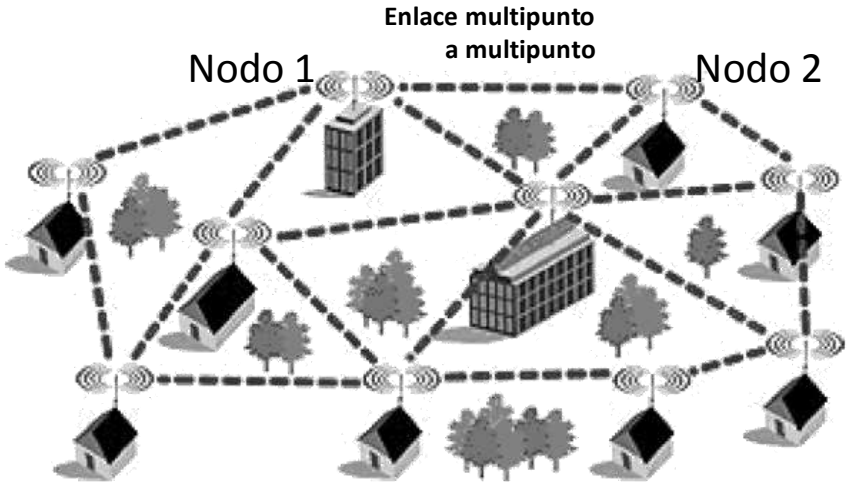


Figura 32

4-Equipos de transmisión y software para la transmisión (protocolo)

Estos temas serán tratados dentro del capítulo de redes para su mejor comprensión.

Capítulo IV
Redes de datos

Se denominan redes teleinformáticas al sistema de transmisión formado por los medios físicos y programas (hardware y software) que permiten la transmisión de informaciones entre computadoras y periféricos. La característica fundamental de las redes es que la comunicación pueda establecerse entre todos los nodos de la red a los efectos de compartir información.

Como ya se dijo en el capítulo I, el concepto de telecomunicaciones se refiere a las comunicaciones a distancia. Además, este mismo concepto involucra la utilización de la electrónica para trasladar las señales a distancia. El tema de redes se suele tratar junto con los temas vinculados a las telecomunicaciones porque involucra muchos de los conceptos de esta última.

Una red de computadoras dentro de un aula destinada a dar un determinado curso no tiene mucho que ver con la transmisión de información a distancia. Pero muchos de los elementos de hardware y software que se utilizan para armar dicha red también son utilizados en las telecomunicaciones y, por ende, son pertenecientes a la disciplina de la teleinformática. Por otro lado, una red que une tres sucursales de una empresa, una ubicada en París, la otra en Tokio y otra en Buenos Aires, ciertamente que se relacionan directamente con la teleinformática. Es decir que a medida que las redes de datos vinculan nodos más alejados, más se relaciona con la teleinformática.

Por lo expuesto, surgió con mucha fuerza el concepto de redes LAN (*Local area network*) o redes de área local y redes WAN (*Wide area network*) o red de área amplia. Obviamente que las redes WAN caben perfectamente en la disciplina de la teleinformática, mientras que las redes LAN se mantienen un poco más alejadas.

Una red LAN podría definirse como una red que une computadoras cercanas (concepto difuso). Son redes que se limitan físicamente a un edificio o a un entorno de no más de una cuadra o de algunas cuadras.

Una red WAN podría definirse como una red que une computadoras o redes LAN que se encuentren a considerable distancia.

Las diferencias entre redes LAN y redes WAN son difusas. Por ejemplo, si la red de computadoras que mencionamos anteriormente, para brindar un curso dentro de un aula, se amplía a otras aulas del mismo edificio, se puede decir que se continúa

dentro del concepto de red LAN. Si se vuelve a ampliar y se incorporan otras aulas de distintos edificios, pero dentro de un mismo predio o cuadra, se podría decir que se continúa dentro del concepto de red LAN; pero sin duda ya el concepto es mucho más difuso y comienza a ingresar dentro del concepto de WAN. Si esa misma red se vuelve a ampliar y se incorpora a edificios que están a varias cuadras de distancia, ya parecería ser que el concepto de LAN desaparece para comenzar a tomar vida el concepto de WAN. O podría considerarse que cada edificio involucrado contiene una red LAN y que las redes LAN de cada edificio están contenidas dentro de una red WAN (Ver Figura 33).

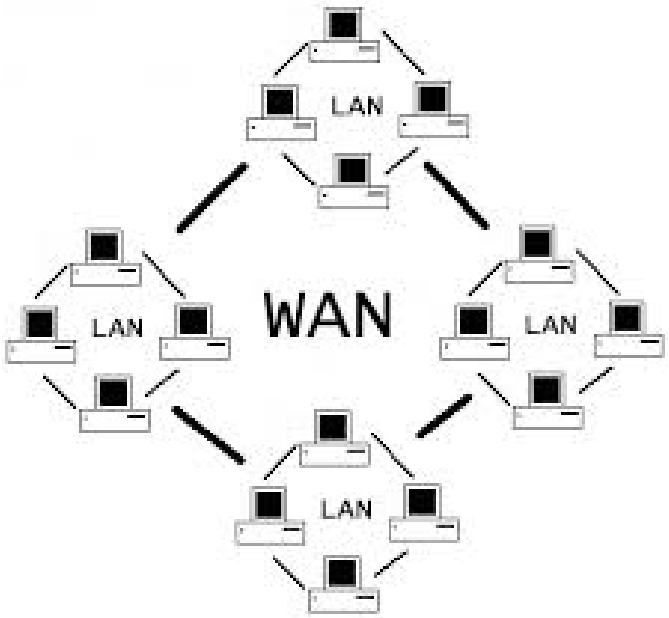


Figura 33

1-Topología de redes

La topología de red la determina, únicamente, la configuración de las conexiones entre nodos. La distancia entre los nodos, las interconexiones físicas, las tasas de transmisión, los tipos de señales y los protocolos no pertenecen a la topología de la red, aunque pueden verse afectados por esta.

1.1-Topología en estrella

Una red en estrella es una red en la cual las estaciones están conectadas directamente a un punto central y todas las comunicaciones se harán necesariamente a través de este. La estación central posee el control total de las estaciones conectadas a ella (Ver Figura 34).

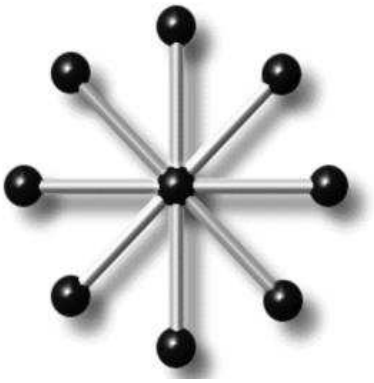


Figura 34

1.2-Topología de anillo

Una red en anillo es una topología de red en la que cada estación tiene una única conexión de entrada y otra de salida. Las computadoras se conectan mediante un canal de comunicación en forma de círculo (Ver Figura 35).

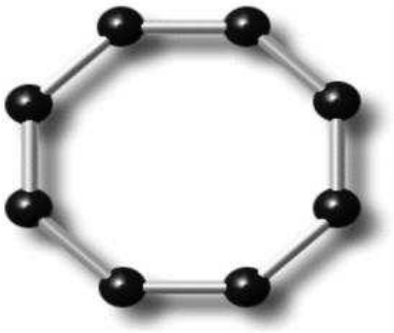


Figura 35

1.3-Topología de bus

Una red en bus es aquella topología que se caracteriza por tener un único canal de comunicaciones (denominado bus, troncal o *backbone*) al cual se conectan los diferentes dispositivos. De esta forma, todos los dispositivos comparten el mismo canal para comunicarse entre sí (Ver Figura 36).

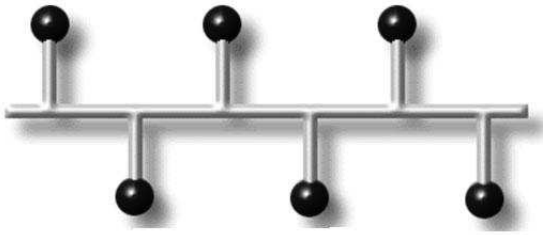


Figura 36

1.4-Topología celular

La topología celular está compuesta por áreas circulares, cada una de las cuales tiene un nodo individual en el centro. La topología celular es un área geográfica dividida en regiones (celdas) para los fines de la tecnología inalámbrica. Esta topología es solo para medios de transmisión no guiados. La transmisión pasa de un nodo al otro por medio de la zona de intercambio donde las células se entrelazan (Ver Figura 37).

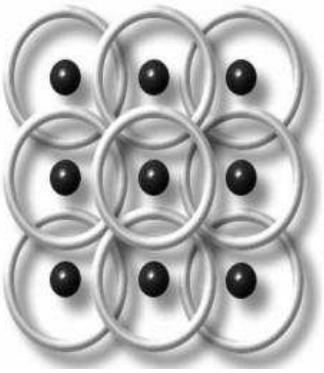


Figura 37

2-Protocolo

Se llama protocolo al conjunto de reglas preestablecidas que regulan la interacción entre computadoras. Estas reglas definen la sintaxis y la sincronización de la comunicación, así como posibles métodos de recuperación de errores. Los protocolos pueden ser implementados por hardware, software o una combinación de ambos.

2.1-Protocolos no orientados y orientados a la conexión

Un protocolo no orientado a la conexión es un método de comunicación en el cual el equipo remitente envía datos sin avisarle al equipo receptor, y este recibe los datos sin enviar una notificación de recepción al remitente. Los datos se envían entonces como bloques. Por el contrario, mediante un protocolo orientado a la conexión, la

transmisión de datos durante una comunicación establecida entre dos máquinas estará controlada. En este esquema, el equipo receptor envía acuses de recepción durante la comunicación, por lo cual el equipo remitente es responsable de controlar que los datos que está enviando lleguen a destino. Los datos se envían entonces como flujo de datos.

3-Red de comunicaciones

Existen dos formas básicas de comunicación en la red: por *switching* o por paquetes.

3.1-*Circuit Switching*

En esta forma, los dos puntos de una conexión mantienen un fluido de información continuo. Los dos extremos de la comunicación mantienen el canal de comunicación en forma exclusiva, mientras que desean estar comunicados. Un ejemplo muy conocido de esta forma es la comunicación telefónica tradicional.

3.2-*Paquet switching* o conmutación de paquetes

A partir de un protocolo que transfiere paquetes conmutados, el intercambio de información entre dos puntos se realizará en bloques de información con un tamaño específico. En el origen, extremo emisor, la información se divide en «paquetes» a los cuales se les indica la dirección del destinatario. Cada paquete contiene, además de los datos, un encabezado con la siguiente información: prioridad (número de orden), direcciones de origen y dirección de destino.

Las ventajas de la conmutación de paquetes son (Ver Figura 38):

a) La eficiencia de la línea es mayor, ya que el canal de comunicación puede ser usado por múltiples comunicaciones simultáneamente. En conmutación de circuitos, la línea se utiliza exclusivamente para una conexión, aunque no haya datos para enviar.

b) Se permiten conexiones entre estaciones de velocidades diferentes, esto es posible ya que los paquetes se irán guardando en cada nodo conforme lleguen.

c) No se bloquean llamadas, ya que todas las conexiones se aceptan, aunque si hay muchas, se producen retardos en la transmisión.

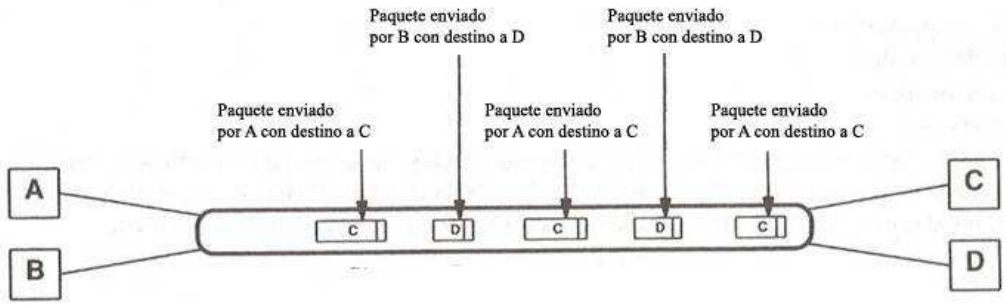


Figura 38

Como se puede ver en la figura 38, tanto el origen A como el B comparten el mismo canal de transmisión. Si, por ejemplo, el origen A envía toda la información en un solo bloque, el origen B debería esperar a que la transmisión de A finalice para recién poder enviar su información. Al conmutar pequeños paquetes de información, se permite que ambas comunicaciones trasmitan simultáneamente por el mismo canal de transmisión.

4-El modelo OSI

El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado OSI (*Open System Interconnection*), es un modelo de red que fue creado por la Organización Internacional para la Estandarización (ISO) en el año 1980. Este modelo es un marco de referencia (*framwork*) para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones. El objetivo de este estándar es facilitar el desarrollo de productos de software y de hardware para la interconexión de computadoras.

Este modelo se basa en separar todas las actividades que se requieren realizar en una comunicación en siete capas. Cada una de las capas tiene un propósito específico y bien diferenciado de la otra. La separación conceptual en capas es una forma de facilitar la incorporación de diversas tecnologías y de múltiples proveedores en un conjunto único de comunicación.

El modelo OSI establece las funciones y capacidades de cada capa, pero el modelo no prescribe cómo debe ser implementada cada capa. El modelo es un marco de referencia que define cada capa y establece los elementos que deben ser

especificados. El foco del modelo está puesto en la interconexión de cada capa, en qué información debe ser pasada de una capa a la otra y en cómo debe hacerse.

Las siete capas definidas en el modelo OSI (Ver Figura 39) son denominadas de acuerdo al propósito de cada una en el proceso de comunicación. Un determinado protocolo puede cubrir las funcionalidades de una capa o más del modelo OSI. Los protocolos de cada capa interactúan con la capa inmediata superior o inmediata inferior para lograr sus objetivos.

Figura 39

La siguiente explicación se desarrolla según la numeración del modelo OSI, pero tal vez sería más comprensible leerla en orden inverso. El modelo es difícil de comprender y lleva tiempo asimilarlo. El lector no debe preocuparse si los conceptos de cada capa le son difusos. Con el correr de la lectura, y sobre todo con la explicación de los protocolos, se podrá ir dando cuenta de cada concepto. Esta primera lectura solo servirá como una pequeña guía. Por otro lado, las compañías que desarrollan tecnología de comunicaciones se guían poniendo más fuerza en su creatividad que en el modelo OSI (si bien tratan de respetarlo para beneficio de todos). El lector debe saber que el principal modelo de comunicación utilizado hoy es el TCP/IP creado por el departamento de Defensa de los EE. UU. diez años antes que el modelo OSI.

4.1-La capa física

Esta capa cumple su función en la transmisión de datos utilizando impulsos eléctricos o señales electromagnéticas. Dentro de esta capa, se deben definir los medios de transmisión posibles, los tipos de conectores para utilizar, los niveles de voltaje, las velocidades y distancias posibles. Se debe definir de qué forma se representan los unos y ceros físicamente.

4.2-Capa de enlace

Esta capa es, fundamentalmente, la encargada de detectar y corregir errores de transmisión, controlar el flujo de datos, construir las tramas (paquete de datos en formato físico), realizar el arbitraje para el acceso a un único canal y encargarse del direccionamiento físico (dirección del hardware). Se debe especificar la topología de red posible.

4.3-Capa de red

La capa de red proporciona conectividad, selecciona la ruta entre dos equipos que pueden estar ubicados en redes totalmente distintas definiendo por qué nodos intermedios pasará la información y es responsable de entregar los paquetes de información. En esta capa se debe especificar cómo se rutean los paquetes de información de una computadora a otra. También, debe controlar la congestión de la ruta para poder derivar la información por otro camino.

4.4-Capa de transporte

Aquí es donde los datos del emisor son reensamblados y controlados. El límite entre la capa de transporte y la capa de sesión puede verse como el límite entre en nivel de aplicación y el de flujo de datos. Mientras que las capas de aplicación, presentación y sesión están relacionadas con la aplicación; las cuatro capas inferiores se encargan del transporte de datos. Las capas de transporte prestan el servicio a las capas de aplicación quitándole a esta última las complejidades técnicas del transporte. Esta capa fragmenta la información en secciones de datos de tamaños que puedan ser transportados por la capa de red y la física. Además, esta capa proporciona confiabilidad a la transmisión utilizando dispositivos de detección y recuperación de errores.

4.5-Capa de sesión

La capa de sesión se encarga de establecer la comunicación (también llamada sesión) entre las computadoras emisora y receptora, y también de finalizarla. La capa de sesión se encarga de establecer elementos de control a las secuencia de datos.

Los protocolos que operan en la capa de sesión pueden ser no orientados a la conexión u orientados a la conexión. Los protocolos orientados a la conexión proporcionan un entorno donde las computadoras conectadas se ponen de acuerdo sobre los parámetros relativos a la creación de los controles para realizar el flujo de los datos. Además, mantienen un diálogo durante la transferencia de los mismos y después terminan de forma simultánea la sesión de transferencia.

4.6-Capa de presentación

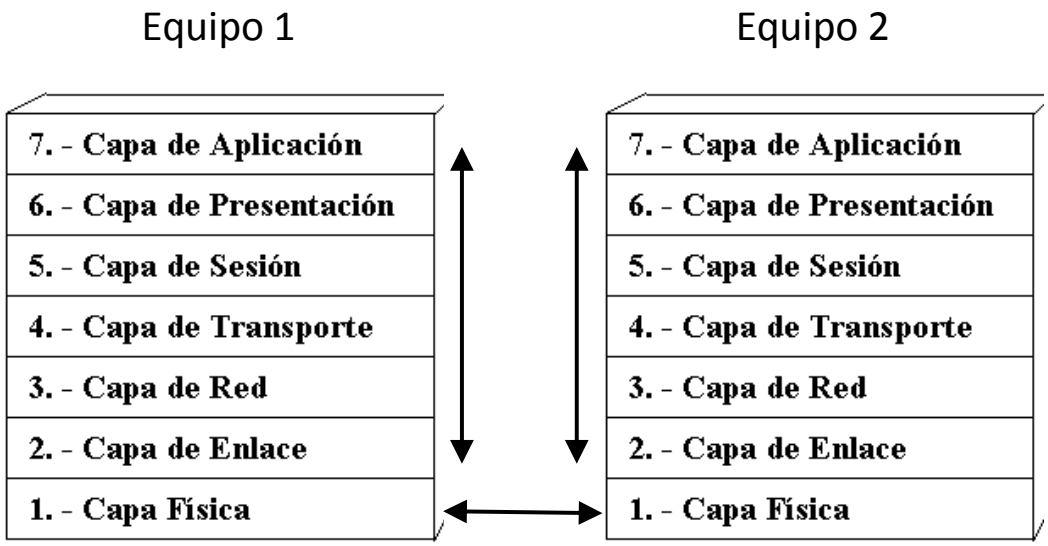
Esta capa toma los datos de la capa de aplicación y los convierte a un formato genérico que puedan leer todas las computadoras. Par ejemplo, los datos en ASCII se traducirán a un formato más básico y genérico.

También, se encarga de cifrar y comprimir los datos para reducir su tamaño y brindar seguridad a la información. El paquete que crea la capa de presentación contiene los datos en la forma con la que viajarán por las restantes capas de la pila OSI (aunque las capas siguientes irán añadiendo elementos al paquete).

4.7-Capa de aplicación

Proporciona la interfaz del usuario con la red. Esta es una capa visible para el usuario.

4.8-Diagrama de interacción entre las capas



5-Encapsulamiento

El intercambio de información entre dos capas OSI o de cualquier otro modelo consiste, básicamente, en que cada capa en el sistema de origen le agrega información de control a los datos, y cada capa en el sistema de destino analiza y quita la información de control de los datos.

Para una mayor comprensión del encapsulamiento, se desarrollará una metáfora asemejando el envío de una carta por correspondencia a una trasmisión (Ver Figura 40).

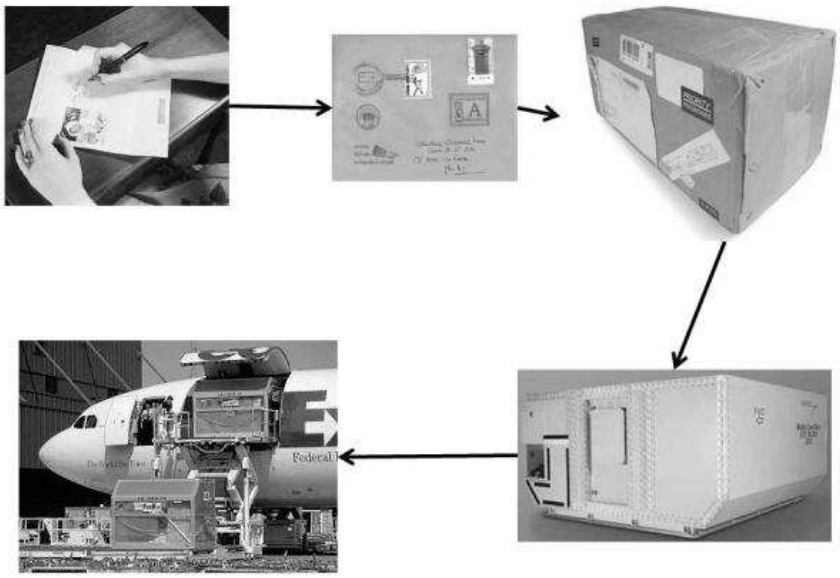


Figura 40

Cuando se redacta una carta, esta se ensobra y en el sobre se coloca la dirección de destino y, por si ocurre algún inconveniente, la dirección de origen o remitente. Luego, en función de la dirección de destino, el sobre se coloca en paquetes más grandes, los cuales son rotulados de forma muy parecida a las de una carta. Después, estos paquetes se ingresan a un conteiner, también rotulado de forma semejante a las anteriores, para que luego el conteiner subido a un avión o a un camión se dirija a un punto de destino donde comienza el proceso inverso hasta llegar a la persona que retira la carta del primer sobre.

El modelo OSI y cualquier otro modelo de comunicación se trabaja de forma semejante. Para enviar información en el modelo OSI (Ver Figura 41), la Capa de Aplicación encapsula los datos agregándole un encabezamiento (*header*), luego la Capa de Presentación hace los mimos, luego la de Sesión, luego la de Transporte. La Capa de Red, además de agregar un encabezado, le agrega una cola como final de paquete, básicamente, para control de integridad. Lo mismo hace la Capa de Enlace que le agrega un encabezado y a la cola un final de trama. La Capa Física se encarga de enviar el flujo de bits de una máquina a la otra.

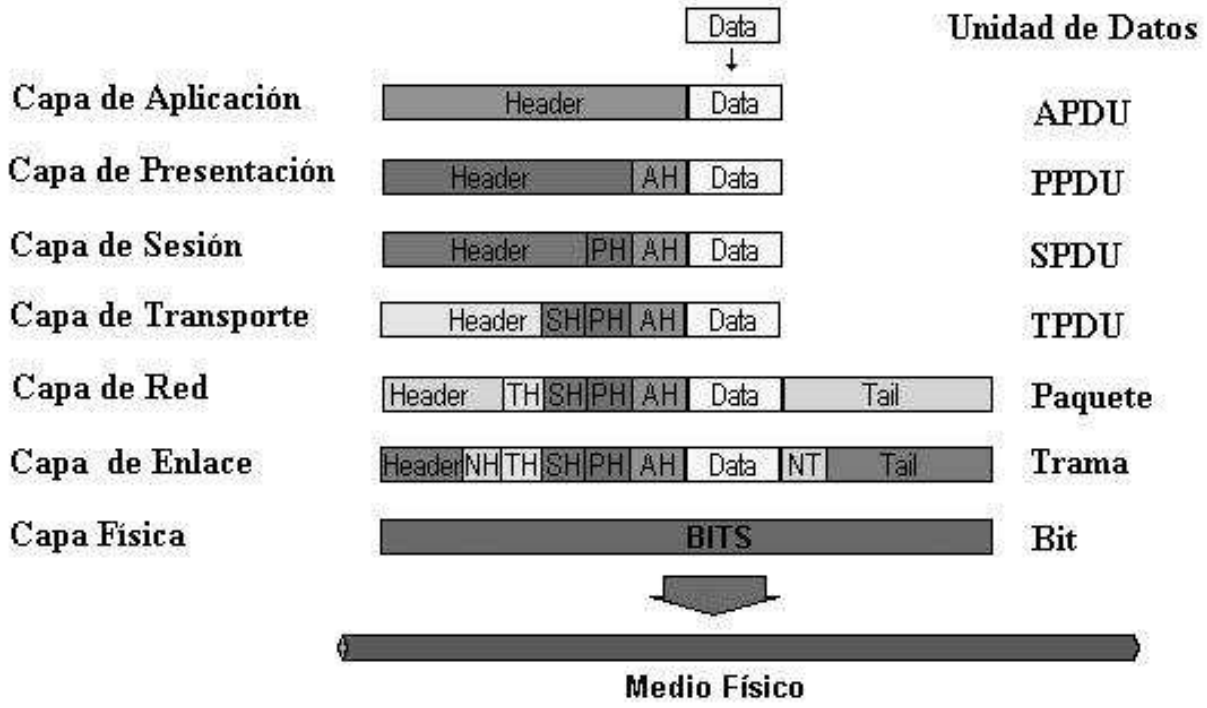


Figura 41

6-El modelo TCP/IP

El modelo OSI, como ya se vio, fue desarrollado por ISO. El modelo TCP/IP fue desarrollado en EE. UU. por el *Defense Advanced Projects Research Agency* (DARPA). Este modelo también es conocido como modelo DoD (*Departament of Defense model*) o como modelo DARPA. Si bien ambos modelos tienen diez años de diferencia, en su creación ambos tienen una considerable similitud. Las capas del modelo TCP/IP son 4:

a) Capa de Aplicación
b) Capa de Transporte
c) Capa de Internet
d) Capa de Acceso a la Red (NAL)

En la figura 42, se puede ver la relación entre el modelo OSI y el modelo TCP/IP.

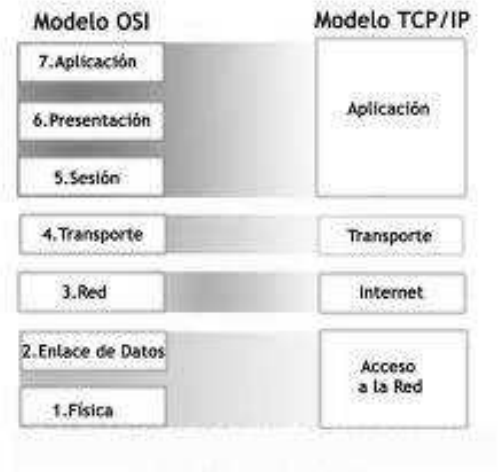


Figura 42

7-Conjunto de protocolos TCP/IP

Los primeros protocolos que desarrolló DARPA fueron TCP e IP. El éxito de estos dos protocolos desarrollados por DARPA desembocó en la construcción de un una serie de protocolos asociados a los que hoy se denomina Conjunto de Protocolos TCP/IP. En la figura 43, se muestran los principales protocolos de esta familia. Existen tantos protocolos en este conjunto que llegan a ser más de 100, aquí veremos los más importantes. TCP e IP es la base de Internet y sirve para enlazar computadoras tanto en redes LAN como sobre redes WAN. Permiten la comunicación entre máquinas con diferentes arquitecturas de hardware y sistemas operativos diferentes. Además, las especificaciones no pertenecen a ningún fabricante, son del dominio público.

CAPAS DEL MODELO TCP/IP	PROTOCOLOS					
Aplicación	TELNET	FTP	SMTP	HTTP		DNS
Transporte	TCP					UDP
Internet	IP					
Acceso a red	Ethernet	DOCSIS	ATM	PPP	Frame Relay	FDDI

Figura 43

Para darle un orden a la explicación del funcionamiento de estos protocolos, primero se narrará un ejemplo del envío de un e-mail de un usuario a otro que atraviesa varias redes y protocolos en su recorrido. El lector no debe preocuparse por lo poco entendible que pueda resultar esta travesía, como ya se explicó, sirve para ordenar la enseñanza de cada componente en particular y para que el lector pueda ubicarse en una serie de figuras que harán de mapa para una mejor comprensión del texto.

El ejemplo es el de un usuario hogareño de la Argentina que desea enviarle un e-mail a un usuario de una gran corporación francesa. A los efectos de una mejor interpretación, el proceso se divide en dos etapas: la primera que va desde el usuario de la Argentina hasta su proveedor de Internet (Ver Figura 44) y la segunda que va desde el proveedor internacional y corporativo de servicios de Internet hasta el usuario de la corporación francesa (Ver Figura 45).

Primera etapa

El usuario de la Argentina envía un e-mail con su aplicación de Microsoft Outlook. Esta funcionalidad implica utilizar un protocolo del nivel de Aplicación denominado SMTP. El protocolo SMTP pasa la información al protocolo TCP, luego este pasa la información al protocolo IP. El usuario está conectado a una red IEEE 802.3 (conocidas comúnmente como Ethernet), que mediante un router-switch y un módem se conectan a Internet. Este usuario utiliza una conexión de cable módem para recibir el servicio de Internet y de televisión por cable. Por lo tanto, el protocolo IP pasa la información a una trama del protocolo IEEE 802.3 (con el que sale la información del computador del usuario de la Argentina) pasa al router-switch, y luego al módem del cable. El módem se conecta con un equipo denominado CMTS (que se encuentra en las oficinas del proveedor del servicio de cable). El proveedor de cable módem utiliza el protocolo DOCSIS (con DOCSIS el e-mail sale del hogar del usuario de la Argentina y llega a los servidores del proveedor de Internet).

El proveedor de Internet por cable tiene una red IEEE 802.3… (se colocan los puntos suspensivos porque los proveedores de Internet utilizan protocolos superiores al estándar básico de IEEE 802.3 como se verá más adelante) a la que están conectados el CMTS (final del protocolo DOCSIS) y el servidor de correo electrónico (SMTP). El CMTS entrega la información a un *switch* IEEE 802.3… que deriva la información hasta el servidor de correo electrónico. En el servidor de correo electrónico, se recibe la trama IEEE 802.3... y se pasa la información al protocolo IP que luego se la pasa al protocolo TCP y que termina entregando la información al servicio de aplicación SMTP.

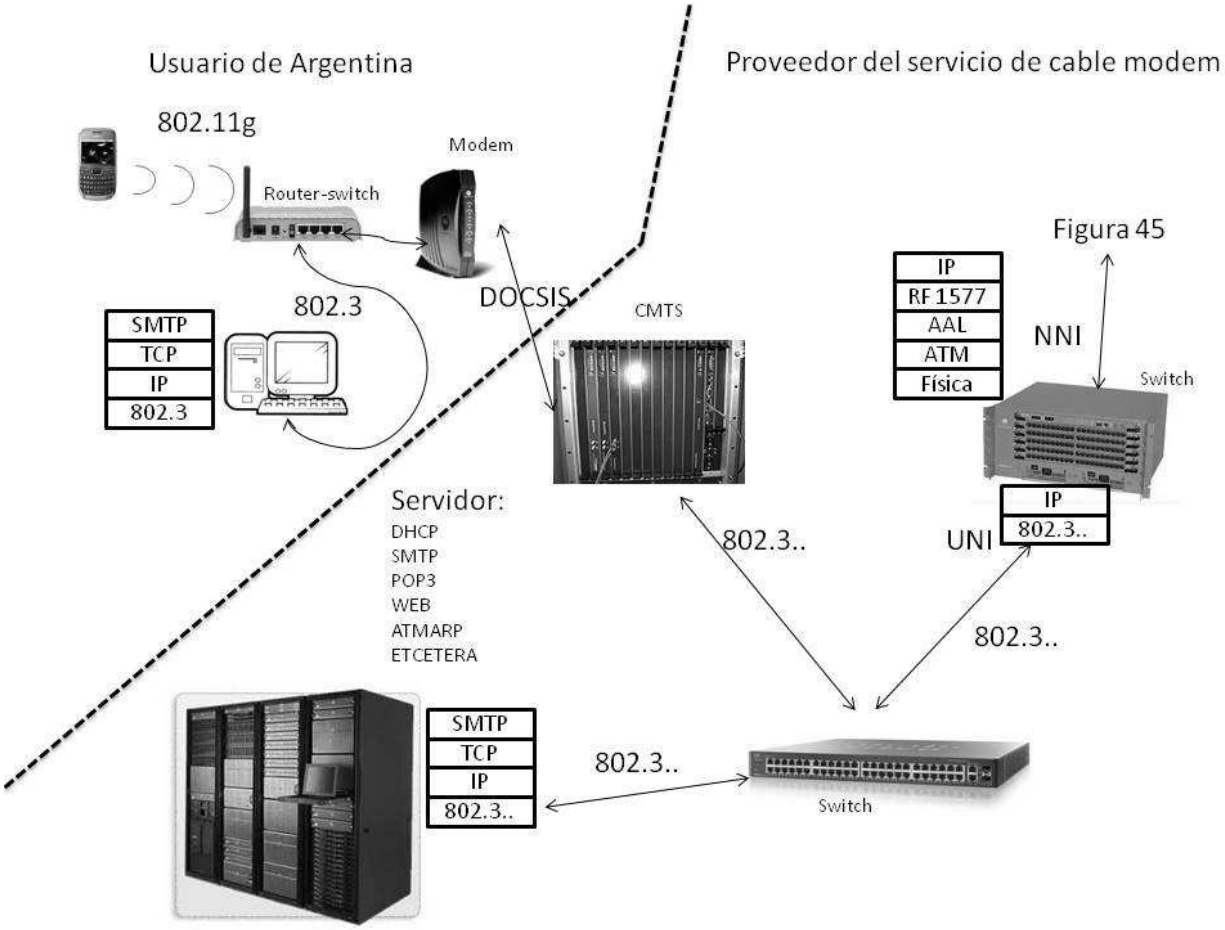


Figura 44

El proveedor de servicios de cable módem se conecta a un proveedor internacional de Internet que utiliza una red ATM para sus comunicaciones internacionales. Este proveedor internacional también presta los servicios de comunicación a la corporación francesa. El e-mail pasa del servicio de aplicación SMTP a TCP, luego a IP, después a IEEE 802.3... que se conecta a una interfaz UNI de un *switch* ATM colocado en las oficinas del proveedor. En el *switch* ATM, la trama IEEE 802.3... es entregada al protocolo IP, este luego se la entrega a la capa del protocolo RF 1577, que luego entrega a la capa AAL del *switch* ATM, que luego pasa a la capa ATM del mismo *switch* y por último a la capa física de ATM para trasmitir las cedas ATM a través de una interfaz NNI al *switch* del proveedor de servicios internacional y corporativo de Internet.

Segunda etapa

Las celdas ATM llegan al *switch* ATM del proveedor internacional y corporativo. Estas celdas son ruteadas por la capa ATM del *switch* ATM y derivadas por una interfaz NNI hasta el *switch* ATM de la corporación francesa. Esta corporación recibe el e-mail en celdas ATM y las pasa a su red corporativa IEEE 802.3 utilizando una interfaz UNI (pasando previamente por las capas ATM, AAL, RF 1577 e IP del *switch* ATM), con la

cual llega al servicio de correo electrónico de la corporación francesa (pasando previamente por las capas IP, TCP y SMTP). El usuario de la corporación francesa usando su Microsoft Outlook (mediante la utilización del protocolo POP3) obtiene del servidor corporativo el e-mail que le enviaron desde la Argentina (previamente se encapsula por TCP, IP y IEEE 802.3 y en la PC del usuario se recompone el e-mail pasando por las capas IP, TCP y POP3).

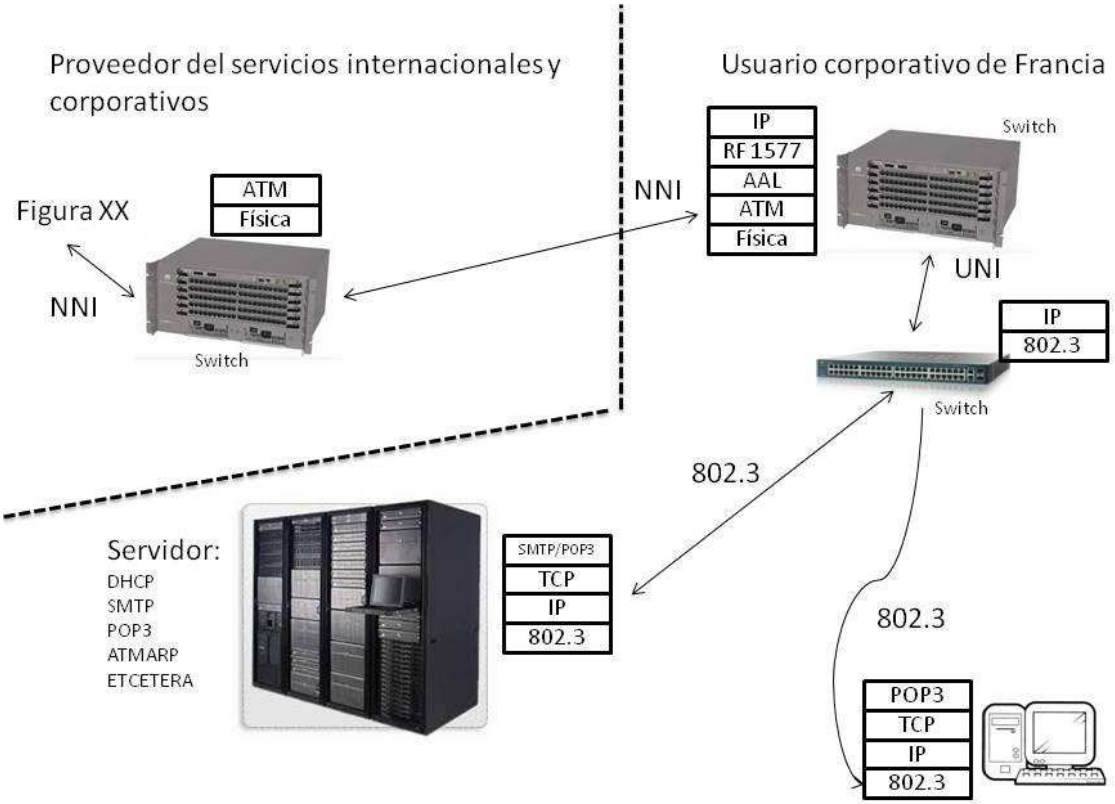


Figura 45

Todo este viaje parece algo imposible de entender, pero al ir recorriendo cada uno de los protocolos todo se irá aclarando. Se le propone al lector que, antes de ingresar a la lectura de cada protocolo en particular, mire estas dos figuras que le permitirán entender mejor el todo sin perderse en las particularidades.

Específicamente, se verán los protocolos siguientes marcados con fondo gris:

CAPAS DEL MODELO TCP/IP	PROTOCOLOS					
Aplicación	TELNET	FTP	SMTP	HTTP		DNS
Transporte	TCP					UDP
Internet	IP					
Acceso a red	Ethernet	DOCSIS	ATM	PPP	Frame Relay	FDDI

7.1-Protocolos de la capa de aplicaciones

Muchos y variados son los protocolos que se utilizan en la capa de aplicaciones. Entre los más conocidos se pueden destacar:

FTP: el Protocolo de Transferencia de Archivos (*File Transfer Protocol*) se utiliza para transferir archivos entre *hosts*.

TELNET (*Telecommunication Network*): este es un protocolo de red que permite viajar a otra máquina para manejarla remotamente como si se estuviera sentados en frente de ella. Telnet solo sirve para acceder en modo terminal, es decir, sin la posibilidad de manejar gráficos, solo caracteres.

HTTP: el Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol*) es el sistema mediante el cual se transfiere información entre los servidores y los clientes. Además, es el método más común de intercambio de información en la *World Wide Web* mediante el cual se transfieren las páginas web a una computadora.

SMTP: el Protocolo para la Transferencia simple de Correo Electrónico (*Simple Mail Transfer Protocol*) se utiliza para el intercambio de mensajes de correo electrónico entre dispositivos.

POP3: El Protocolo de Oficina de Correo (*Post Office Protocol*) se utiliza para obtener los mensajes de correo electrónico almacenados en un servidor de correo electrónico.

Todos estos protocolos utilizan TCP (que se verá más adelante) como capa de transporte y se comunican con ella utilizando las instrucciones primitivas de TCP. Existen otros protocolos de la capa de aplicación que utilizan UDP (se verá más adelante), como el Sistema de Nombres de Dominio DNS (*Domain Naime System*).

Cada uno de estos protocolos tiene una forma muy distinta uno de otro, y dado la variedad, es mejor tomarlos como caja negra sabiendo que emiten y reciben información utilizando como intermediario a TCP.

Se verá con un poco de detalle el SMTP, ya que es la aplicación que se utilizó para el ejemplo.

El servicio de correo electrónico se realiza mediante la modalidad de procesamiento cliente-servidor. El proceso de envío de un e-mail se desarrolla en cuatro etapas, en este caso Andrés Fernández envía un e-mail desde su casilla de correo electrónico a andres_fernandez@atlantamail.com.ar con destino a la casilla de correo electrónico carlos_rodriguez@siliconelectronico.com.ar (Ver Figura 46).

I. Andrés Fernández envía un e-mail utilizando un cliente de correo electrónico (para lo cual se utiliza el protocolo de SMTP) que puede ser por ejemplo Microsoft Outlook.

II. Este mail primero llega al servidor de correo electrónico del proveedor del servicio de correo que utiliza Andrés Fernández y que se identifica por el dominio de la dirección de correo saliente (atlantamail.com.ar). El servidor de correo electrónico de atlantamail.com.ar también utiliza el protocolo SMTP para recibir el e-mail. El proveedor del servicio de correo electrónico puede ser un área de la misma empresa, no necesariamente tiene que ser un proveedor externo.

III. Si el destinatario del e-mail también es cliente del mismo proveedor, el servidor de correo electrónico dejara el e-mail en la casilla de correo que corresponde al destinatario (que no es este caso). Si el destinatario es de otro proveedor de servicio de correo electrónico, el servidor envía el e-mail hacia el dominio de la dirección del destinatario del e-mail (por ejemplo, en la dirección carlos_rodriguez@siliconelectronico.com.ar, el dominio es siliconelectronico.com.ar).

IV. El servicio de correo electrónico del dominio siliconelectronico.com.ar recibe el e-mail y lo guarda en la casilla de carlos-rodriguez (utilizando el protocolo SMTP).

V. Por último, el destinatario, desde su cliente de correo electrónico, retira el e-mail de su casilla de correo; pero en este caso debe utilizar el protocolo POP3.

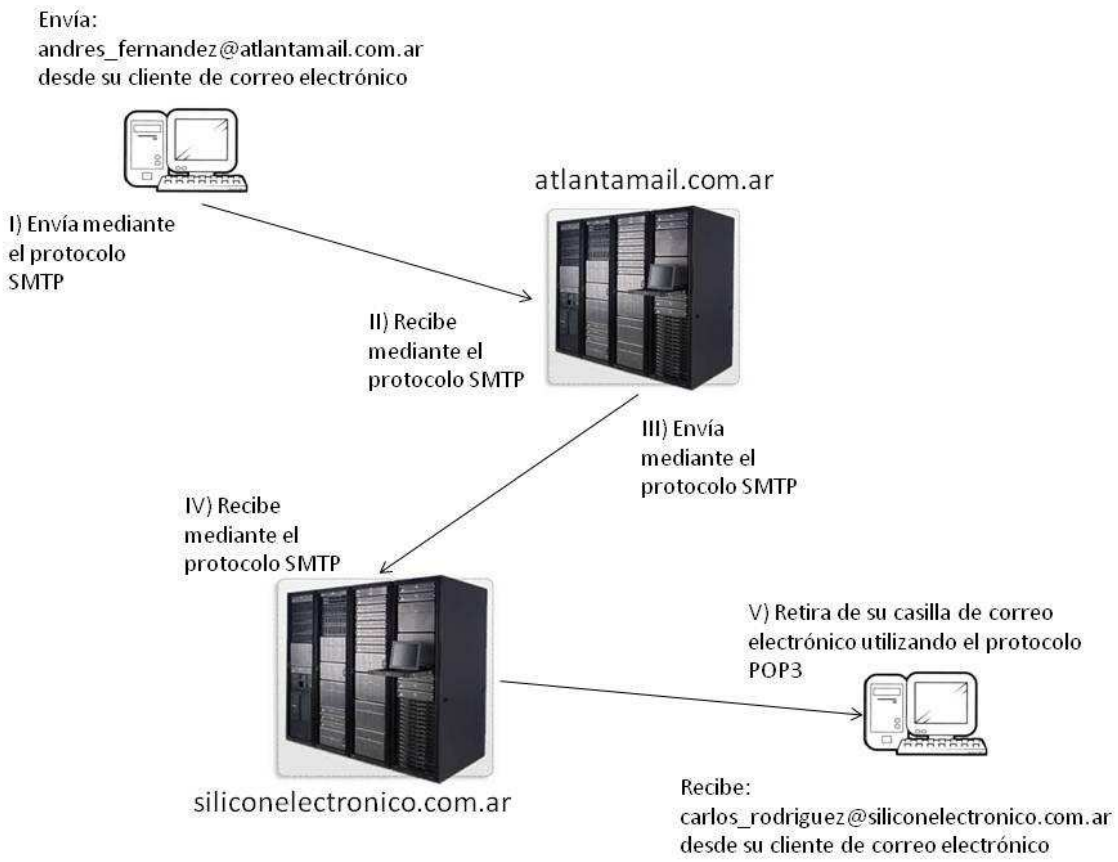


Figura 46

La transferencia de un e-mail envuelve una serie de comandos SMTP. Cada uno de estos comandos se operan mediante una cadena de caracteres ASCII que pueden estar formados por: tres caracteres numéricos o por texto o por ambos. Estos comandos se envían a la red utilizando las primitivas. Las primitivas son programas que funcionan como si fueran enchufes a través de los cuales las aplicaciones se conectan con TCP.

A continuación se mostrarán algunos comandos de la interacción del cliente de correo electrónico y del servidor de correo electrónico con un gran nivel de simplificación.

El SMTP cuando inicia la conexión emite un comando denominado «HELO - nombre del dominio del servidor», este *string* de caracteres se envía con el comando Send de TCP. A este comando debería seguir una respuesta del servidor que debería ser su nombre de dominio. Suponiendo que se recibe la respuesta sin errores, el siguiente comando debería ser MAIL. El comando MAIL indica al servidor el inicio de un mensaje de correo y le indica además que quien lo envía es el remitente del mensaje. Si está todo bien, la respuesta del servidor debería ser 250 OK. El siguiente

mensaje debería ser RECEPT TO –destinatario del correo. Si todo está bien, la respuesta debería ser 250 OK. El siguiente mensaje debería ser DATA, para notificar al receptor que a continuación se envían los contenidos del mensaje, etcétera.

Como la intención aquí no es desarrollar el protocolo SMTP, no se extenderá más esta pequeña introducción. El objetivo es mostrar que los comandos SMTP se pasan al protocolo TCP utilizando las primitivas que se pueden observar en la figura 48.

7.2-Protocolos de transporte

El nivel de transporte o capa de transporte es el segundo nivel del modelo TCP/IP. Es el encargado de la transferencia libre de errores de los datos entre el emisor y el receptor (aunque no estén directamente conectados), así como de mantener el flujo de la red. La tarea de esta capa es proporcionar un transporte de datos confiable y económico de la máquina de origen a la máquina destino, independientemente de la red o redes físicas que deban atravesarse.

Para permitir que las aplicaciones accedan al servicio de transporte, la capa de transporte debe proporcionar algunas operaciones a los programas de aplicación, es decir, una interfaz del servicio de transporte a las que se llaman primitivas.

7.2.1-Protocolo TCP

El Protocolo de Control de Transmisión TCP (*Transmission Control Protocol*) es un protocolo orientado a la conexión (está documentado en la RFC 793). Es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. Muchos programas dentro de una red de datos pueden usar este protocolo para generar conexiones entre sí. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. Además, proporciona un mecanismo para diferenciar aplicaciones dentro de una misma máquina, a través del concepto de puerto.

En el nivel de transporte, los paquetes de bits que constituyen las unidades de datos del protocolo TCP se llaman «segmentos». Es decir que este protocolo toma los datos que llegan del nivel de aplicación (*string* de caracteres), los separa en pequeños segmentos, le agrega una cabecera a cada uno de ellos y se los entrega al protocolo IP. Este protocolo también es el encargado de realizar el trabajo inverso, es decir que toma los paquetes que le entrega el protocolo IP y reconstruye la cadena de datos para entregarla al nivel de aplicación. Para esta tarea debe realizar un ordenamiento de los paquetes entregados por IP. TCP permite multiplexar datos de diferentes fuentes (por ejemplo, aplicaciones) y es quien se encarga de abrir y cerrar una conexión.

El tamaño del segmento de TCP es variable, pero se encuentra limitado por la cantidad de bytes que soporta el paquete IP (65.535 bytes).

7.2.1.1-Puerto

TCP, para cumplir con su objetivo, necesita saber a qué aplicación de la máquina de destino le está enviando la información y de qué aplicación de la máquina de envío sale la información. Este objetivo lo logra utilizando un número al que se denomina número de puerto (este es un concepto de puerto lógico, que se asocia con una aplicación y que no debe confundirse con los puertos físicos). El rango de números posibles de puertos está entre el 1 y el 65.535. Del 1 al 1023 ya están reservados por ICANN y el resto son de libre uso. Por ejemplo, entre los más conocidos, se pueden nombrar: FTP/21, HTTP/80, Telnet/23, SMTP/25, POP3/110.

7.2.1.2-*Socket*

A la unión del número de IP, gestionado por el protocolo IP, y el número de puerto, gestionado por el protocolo TCP, se denomina *socket*. Por lo tanto, un número de *socket* identifica a una determinada aplicación en un específico *host* (tanto de la aplicación/*host* de origen como de la aplicación/*host* de destino).

7.2.1.3-El segmento TCP

En la figura 47, se puede ver el diseño de la cabecera del segmento TCP, se explicarán los más significativos.

Puerto de origen: indica el puerto emisor.
Puerto de destino: indica el puerto receptor.
Número de secuencia: TCP garantiza que la información sea recibida en orden. Para ello, cada segmento enviado tiene un número de secuencia. Este número se inicia con un valor aleatorio (es decir que el primer segmento que se envíe de una comunicación recién iniciada será un número aleatorio). En el siguiente segmento, este número irá incrementándose según la cantidad de bytes enviados en el segmento anterior. Por ejemplo, si un segmento tiene un número de secuencia **x** y contiene **k** bytes, el número de secuencia del siguiente paquete emitido será **x** + **k**. Concretamente este número va contando la cantidad de bytes de datos que envía un *host* al otro (por cada *socket*).
Número de acuse de recibo: cuando el receptor reciba el segmento, deberá enviar un número de recibo que será igual al «número de secuencia» más la cantidad de bytes recibidos (este será el «número de secuencia» del segmento siguiente). De esta forma, TCP sabe si el destino ha recibido la información completa.
Longitud de cabecera: se indica en cantidad de palabras (la palabra es igual a 4 bytes) de la cabecera.

Reservado: es un campo que no está en uso y se reserva para alguna necesidad futura, su longitud es de 6 bits.
URG: si este indicador está fijado en 1, el paquete se debe procesar en forma urgente.
ACK: si este indicador está fijado en 1, el paquete es un acuse de recibo.
SYN: este indicador fijado en 1 identifica al segmento enviado como un segmento de sincronismo, a continuación se verá cómo es esta función.
FIN: si este indicador está fijado en 1, se interrumpe la conexión.

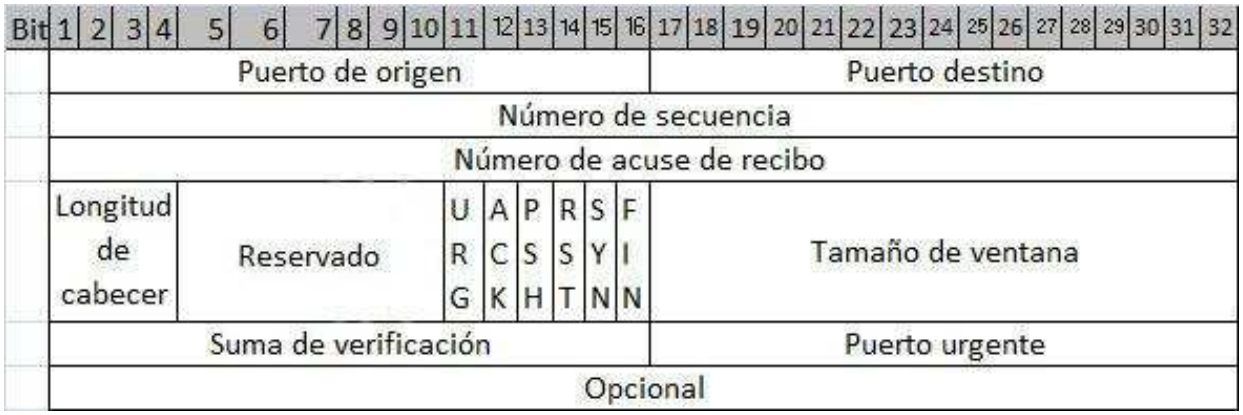


Figura 47

7.2.1.4-Esquema de funcionamiento

Cuando se crea un segmento, simultáneamente, se genera un número de secuencia (como se explicó anteriormente). La máquina receptora, luego de recibir un segmento, emite un segmento con el «número de acuse de recibo» (creado como se vio anteriormente), se pone en 1 el indicador ACK y se envía el segmento al remitente (*socket* de origen).

El protocolo TCP tiene un temporizador que luego de transcurrido un lapso de tiempo, si el *host* emisor no recibe el acuse de recibo, vuelve a enviar el segmento. Si la falta de aviso de retorno fue por un retraso en la llegada del segmento al receptor, lo que ocurrirá es que el *host* receptor recibirá varias veces el mismo segmento. Pero utilizando el «número de secuencia», el protocolo TCP del receptor sabrá que se trata de iguales segmentos y solamente tomará uno de ellos.

El diálogo entre las dos máquinas se inicia de la siguiente forma:

I. Un *host* debe tener abierta una sesión de TCP como servidor (*host* receptor). Este *host* está en espera de recibir mensajes, en estado pasivo.

II. La sesión en el *host* del cliente (*host* que envía) debe estar abierta en forma activa.

III. El intercambio comienza con una sincronización de ambos *hosts* (cliente y servidor) en tres pasos.

IV. El primer paso es cuando el *host* cliente transmite un segmento con el indicador SYN fijado en 1 (esto significa que es un segmento de sincronización). En este mismo segmento, se informa el «número de secuencia inicial» del *host* cliente.

V. Cuando el *host* servidor recibe el segmento, emite un acuse de recibo con el indicador ACK fijado en 1 (esto significa que este segmento, además de ser un acuse de recibo, es un sincronizador). Este mismo segmento incluye el «número de secuencia» inicial del *host* servidor. El «número de acuse de recibo» de este segmento es el «número de sincronización» del *host* cliente más 1.

VI. En el tercer paso, el *host* envía un acuse de recibo (con ACK fijado en 1), con SYN fijado en 0 (este ya no es un segmento de sincronización) y el «número de acuse de recibo» es el «número de secuencia» inicial del *host* servidor más 1.

Luego de que estos pasos se dieron exitosamente, se puede iniciar la comunicación.

Para terminar una conexión, se siguen los siguientes pasos:

I. Uno de los *host* envía un segmento con FIN fijado en 1.

II. El otro *host* envía un acuse de recibo con FIN fijado en 1

III. Luego quien mando el acuse de recibo debe mandar un segmento con FIN fijado en 1 y esperar el acuse de recibo para dar, definitivamente, por cerrada la conexión. Este doble juego de envío de fin de conexión y acuse de recibo es porque uno de los *hosts* puede querer seguir enviando información.

7.2.1.5-Primitivas de TCP

Para que las aplicaciones puedan utilizar el protocolo TCP, este provee de unas interfaces que permiten conectar a la aplicación con TCP. Estas interfaces presentan una serie de instrucciones (primitivas) que pueden ser utilizadas por la capa de aplicaciones para manejar la capa de transporte, por ejemplo:

PRIMITIVA	TIPO	CLIENTE / SERVIDOR	PARÁMETROS
Apertura pasiva no específica UNSPECIFIED PASSIVE_OPEN	Petición	S	Puerta de origen, temporizador, acción en caso de expiración de temporizador, precedencia, rango de seguridad.
Apertura completa pasiva FULL_PASSIVE_OPEN	Petición	S	Puerta de origen, puerta de destino, dirección de destino, temporizador, acción en caso de expiración de temporizador, precedencia, rango de seguridad.
Apertura activa ACTIVE_OPEN	Petición	C	Puerta de origen, puerta de destino, dirección de destino, temporizador, acción en caso de expiración de temporizador, precedencia, rango de seguridad.
Apertura activa con datos incluidos ACTIVE_OPEN_WITH DATA	Petición	C	Puerta de origen, puerta de destino, dirección de destino, temporizador, acción en caso de expiración de temporizador, precedencia, rango de seguridad.
identificativo de la apertura OPEN_ID	Respuesta (local)	C	Nombre de conexión local, puerta origen, puerta destino, dirección destino.
Apertura exitosa OPEN_SUCCESS	Confirmación	C	Nombre de conexión local.
Apertura fallada OPEN_FAILURE	Confirmación	C	Nombre de conexión local.
Transmitir SEND	Petición	C/S	Nombre de conexión local, datos, longitud de datos, flag de push, flag de urgente, temporizador, acción en caso de expiración de temporizador.
Entregar DELIVER	Indicación	C/S	Nombre de conexión local, datos, longitud de datos, flag de urgente.
Asignar ALLOCATE	Petición	C/S	Nombre de conexión local.
Cerrar CLOSE	Petición	C/S	Nombre de conexión local.
Cerrando CLOSING	Indicación	C/S	Nombre de conexión local, código de motivo.
Terminar TERMINATE	Confirmación	C/S	Nombre de conexión local.
Abortar ABORT	Petición	C/S	Nombre de conexión local.
Estado STATUS	Petición	C/S	Nombre de conexión local.
Respuesta de estado STATUS_RESPONSE	Petición	C/S	Nombre de conexión local, puerta origen, dirección de origen, puerta de destino, ventana de recepción, ventana de transmisión, esperando ACK, esperando recepción, urgente, precedencia, rango de seguridad, temporizador.
ERROR	Indicación	C/S	Nombre de conexión local, código de motivo.

Figura 48

7.3-Protocolo UDP

User Datagram Protocol (UDP) es un protocolo no orientado a conexión. Está definido por la RFC 768. Se utiliza en reemplazo de TCP cuando la velocidad es importante y la confiabilidad no representa una inquietud. No necesita ni apertura ni cierre de conexión, tampoco realiza confirmación de recepción ni control de datos enviados. Es usado, básicamente, por los protocolos DNS, DHCP y todos los productos de audio y video en tiempo real. Cualquier garantía o confiabilidad que se le quiera dar a la transmisión debe hacerse en el nivel de la capa de aplicación. La cabecera del segmento UDP es muy simple, consta de cuatro campos de 16 bits cada uno donde se informa en uno el puerto de origen (que es un dato opcional), luego el puerto de destino, la longitud del mensaje y la suma de verificación que también es un dato opcional.

7.4-Protocolo IP

Internet Protocol (en español, Protocolo de Internet, IP) es un protocolo de comunicación de datos digitales que se ubica en la capa de red según el modelo internacional OSI, y en la capa de Internet, según el modelo TCP/IP. Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos mediante un protocolo no orientado a conexión que transfiere paquetes conmutados a través de distintas redes físicas. El protocolo IP no transporta bits de un lugar a otro, de eso se encargan los protocolos como IEEE 802.3, ATM, Frame Relay, DOCSIS, etcétera. Estos protocolos pertenecen al nivel de acceso a red en el modelo TCP/IP.

7.4.1-Dirección IP

Existen dos versiones de protocolos IP: la versión IPv4 y la IPv6. El IPv6 es una versión diseñada para reemplazar a IPv4. El protocolo IPv4 es el que actualmente utiliza, mayoritariamente, Internet. Lentamente las redes de Internet están teniendo que pasar a IPv6 para superar las limitaciones al crecimiento que significa IPv4. La limitación principal de IPv4 es la cantidad de dispositivos que pueden conectarse a la red. IPv4 permite conectar 4.294.967.296 (2^{32}) de dispositivos, mientras que IPv6 permite conectar 340.282.366.920.938.463.463.374.607.431.768.211.456 (2^{128} o 340 sestillones de direcciones).

7.4.1.1-IPv4

Una computadora que desea comunicarse con otra computadora dentro de una red que utiliza el protocolo IP debe saber el número IP de la computadora de destino y el número IP propio. Cada computadora dentro de una red debe tener una dirección única. Esto es muy similar a cuando una persona quiere llamar por teléfono a otra, primero debe saber el número telefónico de la persona que desea llamar. Fíjese el lector que hasta aquí no se ha hablado de direcciones. Ni los protocolos del nivel de aplicación ni TCP ni UDP hasta aquí se han preocupado por una dirección de destino.

Las direcciones IPv4 están formadas por cuatro octetos binarios. Por ejemplo, una dirección IPv4 puede ser la siguiente: 11000000101010000000000100001000. Las direcciones IPv4 se muestran con un punto que separa cada octeto: 11000000.10101000.00000001.00001000. Para que esta dirección sea más comprensible para los humanos, la computadora las muestra en decimal, por ejemplo: 192.168.1.8, que es la expresión decimal del octeto anteriormente descripto, por ejemplo: 192 es igual al número binario 11000000.

La organización responsable del otorgamiento de las direcciones IP es la Corporación de Internet para la Asignación de Nombres y Números (en inglés, *Internet Corporation for Assigned Names and Numbers*, ICANN).

El protocolo IP intercambia información conmutando paquetes (también llamados datagramas). Cada paquete IP contiene en su cabecera la dirección IP de destino y la dirección IP de origen. Estas direcciones IP están construidas de esta forma para poder conectar computadoras de distintas redes y que estas se localicen rápidamente.

7.4.1.1.1-Jerarquía de las direcciones IPv4

Las direcciones IP tienen un orden jerárquico que facilita la ubicación de una dirección concreta dentro de toda la red (Ver Figura 49).

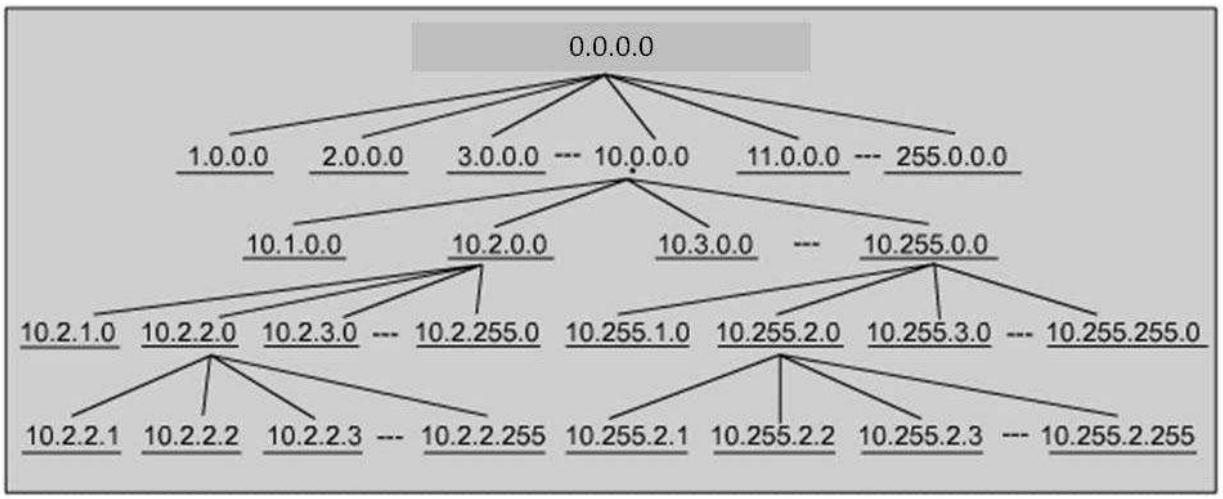


Figura 49

El nodo superior de la jerarquía está representado por la dirección 0.0.0.0 que no se utiliza. El primer grupo de octetos representa la primera familia del árbol, el segundo octeto la segunda familia del árbol, el tercer octeto representa la tercera familia del árbol y el cuarto octeto sería como las hojas del árbol.

Para el otorgamiento y la administración de las direcciones IP, la ICANN establece cuatro clases de direcciones: A, B, C y D. Esta clase se identifica por el número del primer octeto.

Las direcciones clase A se identifican porque el primer octeto puede estar en el rango del 1 al 126. Cada número de este rango identifica una red (grupo). Los siguientes tres octetos pueden ser utilizados libremente por cada organización. La organización que posee una dirección IP clase A podrá asignar hasta un máximo de 16.777.214 *hosts* libremente (cabe destacar que al momento de escribirse este libro las direcciones IP ya han sido todas otorgadas).

Las direcciones clase B se identifican porque el primer octeto puede estar en el rango del 128 al 191. Cada número de este rango identifica una red. Poseer una dirección IP clase B significa obtener un número de red del primero octeto (entre 128 y 191) y un número de red del segundo octeto entre 0 y 254. Quien posee una dirección IP clase B puede asignar hasta un máximo de 65.534 *hosts* libremente.

Las direcciones clase C se identifican porque el primer octeto puede estar en el rango del 192 al 223. Cada número de este rango identifica una red. Poseer una dirección IP clase C significa obtener un número de red del primero octeto (entre 128 y 191), un número de red del segundo octeto (entre 0 y 254) y un número de octeto de la tercera red (entre 0 y 254). Quien posee una dirección IP clase C puede asignar hasta un máximo de 254 *hosts* libremente.

Las direcciones clase D se identifican porque el primer octeto puede estar en el rango del 224 al 239. Estas direcciones IP son utilizadas únicamente como *multicast*, por lo que la misma IP puede ser asignada a múltiples usuarios. Estas IP son utilizadas únicamente para recibir información. Todos los usuarios con la misma IP reciben la misma información.

En las figuras 50 y 51, se representa esquemáticamente lo antedicho.

Clase A	Red	Host		
Octeto	1	2	3	4

Clase B	Red		Host	
Octeto	1	2	3	4

Clase C	Red			Host
Octeto	1	2	3	4

Clase D	Host			
Octeto	1	2	3	4

Figura 50

Clase de dirección	Intervalo de dirección del primer octeto	Número de bits en la dirección de red	Número de redes	Número de hosts por red
Clase A	0-127	8	126	16,777,216
Clase B	128-191	16	16,384	65,536
Clase C	192-223	24	2,097,152	254
Clase D	224-239	28	No es aplicable	No es aplicable

Figura 51

7.3.1.1.2-Subredes

El protocolo IP permite crear subredes de dos formas:

a- Utilizando los octetos libres

Dentro de una LAN, cuando el poseedor de una IP clase A desea crear subredes, puede utilizar el segundo y tercer octeto de la dirección IP a tal fin.

Ejemplo: una IP clase A como la siguiente 124.___.___.___

En el caso de poseer una IP clase B, puede utilizar el tercer octeto.

Ejemplo: una IP clase B con la siguiente 129.35.___.___

En una clase C, ya no quedan octetos para construir subredes.

b- Utilizando la funcionalidad de máscaras contemplada dentro del protocolo IP

La máscara permite destinar una parte de un octeto a numerar una red y otra parte a ser utilizada como numeración del *host*. Esta técnica requiere de asociar un número de máscara (similar a los números IP) al número IP asignado.

Si un usuario obtiene una IP clase C, ya tiene tres octetos que no puede modificar, pero le queda un octeto libre.

Por ejemplo: 198.1.1.__

Este número IP en binario es: 11000110.00000001.00000001._____

Como se puede ver en el ejemplo, quedan 8 bit para utilizar. En esta técnica con los unos (1) se marcarán las posiciones para la red y con los ceros (0) las posiciones para el número de *host*. Para calcular el número de subredes y de *hosts*, se usa la fórmula de 2 elevado a la n donde n es la cantidad de bit en unos para redes y en ceros para *host*. En ambos casos, hay que restarle dos porque se necesita una dirección para *broadcast* y otra para la red.

De esta forma las alternativas para el último octeto serían:

11111110 lo que permitiría tener 128 subredes con 2 *hosts* asociado a cada red, pero que en la práctica no es posible implementar.
11111100 lo que permitiría tener 64 subredes con 4 *hosts* asociados a cada red.
11111000 lo que permitiría tener 32 subredes con 8 *hosts* asociados a cada red.
11110000 lo que permitiría tener 16 subredes con 16 *hosts* asociados a cada red.
11100000 lo que permitiría tener 8 subredes con 32 *hosts* asociados a cada red.
11000000 lo que permitiría tener 4 subredes con 64 *hosts* asociados a cada red.
10000000 lo que permitiría tener 2 subredes con 128 *hosts* asociados.

Como el lector puede observar, no por casualidad el resultado de multiplicar las redes por los *hosts* da como resultado 256 que son el total de combinaciones posibles con 8 bits.

Por ejemplo, si se tiene asignada una IP clase C 200.3.25.0 y se pretendiera armar 8 subredes, la distribución de direcciones IP sería la siguiente:

Red	Rango IP desde	Rango IP hasta	Broadcast
200.3.25.0	200.3.25.1	200.3.25.30	200.3.25.31
200.3.25.32	200.3.25.33	200.3.25.62	200.3.25.63
200.3.25.64	200.3.25.65	200.3.25.94	200.3.25.95
200.3.25.96	200.3.25.97	200.3.25.126	200.3.25.127
200.3.25.128	200.3.25.129	200.3.25.158	200.3.25.159
200.3.25.160	200.3.25.161	200.3.25.190	200.3.25.191
200.3.25.192	200.3.25.193	200.3.25.222	200.3.25.223
200.3.25.224	200.3.25.225	200.3.25.254	200.3.25.255

Figura 52

La cantidad de *host* por redes surge de dividir 256 por 8.

Siempre se debe reservar una dirección para la red y una dirección de *broadcast*.

La dirección de broadcast sirve para mandar la misma información a todas las IP de la misma subred.

Si bien la resolución técnica de la aplicación de máscaras no es del alcance de este libro, vale la pena agregar que para que la máscara (número asociable a la dirección IP asignada) funcione como el caso explicado anteriormente (IP 200.3.25.0), debería ser 255.255.255.224. El 224 en binario es el numero 11100000, si el lector vuelve a mirar las alternativas de los octetos, verá que para 8 redes es necesario usar tres unos y para 32 *hosts* es necesario usar 5 ceros. Las alternativas de valor para el último octeto de la máscara son: 254, 252, 248, 240, 224, 192, 128 que va desde 128 subredes hasta dos subredes. El 255 (que son todos unos) indica que el valor que se toma para ese octeto es el de la IP asignada.

7.4.1.1.3-El datagrama

El datagrama, también llamado paquete, es la unidad primaria de agrupamiento de bits. El lector recordará que para poder utilizar un canal de comunicación compartido es necesario que quienes estén utilizando el medio seccionen la información en pequeñas (datagramas) para que estos se puedan ir intercalando (conmutando) con la finalidad de hacer posible la compartición del medio.

El datagrama debe llegar desde la computadora de origen hasta la computadora de destino, cómo lograr esto es un problema de los protocolos del nivel de acceso a la red. El datagrama IP puede ser trasportado por varios protocolos de acceso a red hasta que llega a destino.

La longitud máxima de un datagrama es de 65.536 octetos (bytes). Su cabecera consta de 5 palabras de 32 bits cada una (Ver Figura 53) y sus principales campos son:

<table>
<tr><th>0-3 bits</th><th>04-07 bits</th><th>08-15 bits</th><th>16-18 bits</th><th>19-31 bits</th></tr>
<tr><td>Versión</td><td>Tamaño Cabecera</td><td>Tipo de Servicio</td><td colspan="2">Longitud Total</td></tr>
<tr><td colspan="3">Identificador</td><td>Flags</td><td>Posición de Fragmento</td></tr>
<tr><td colspan="2">Tiempo de Vida</td><td>Protocolo</td><td colspan="2">Suma de Control de Cabecera</td></tr>
<tr><td colspan="5">Dirección IP de Origen</td></tr>
<tr><td colspan="5">Dirección IP de Destino</td></tr>
<tr><td colspan="4">Opciones</td><td>Relleno</td></tr>
</table>

Figura 53

Tamaño de la cabecera: es un dato importante porque la cabecera puede ser de longitud variable debido a que en ella hay datos opcionales. El valor mínimo es de 160 bits.

Tipo de servicio: es un indicador para establecer la calidad de servicio, es como un identificador de prioridad.

Longitud total: se refiere al total de bits que contiene el datagrama. El datagrama es de longitud variable dependiendo de la cantidad de información que transporte.

Tiempo de vida: contiene un número que es disminuido en uno cada vez que pasa por un nodo (unión de dos redes). Cuando llega a cero, el datagrama se descarta. Esto es para evitar redundancias cíclicas (que el paquete circule en la red sin lograr llegar al destino) y que de esta forma se degrade el funcionamiento de la red.

Dirección IP origen: dirección IP de origen del datagrama.

Dirección IP de destino: dirección IP de destino del datagrama.

7.4.1.1.4-Router IP

Un router IP –también conocido como enrutador o encaminador de paquetes– es un dispositivo que envía o encamina datagramas de datos de una red a otra, es decir, interconecta subredes (Ver Figura 54). En este caso, por tratarse de dos redes ubicadas en el mismo lugar físico, puede utilizarse un solo router.

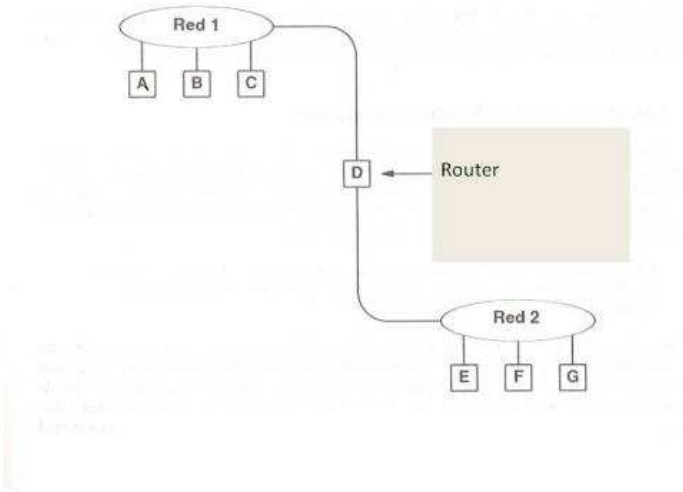


Figura 54

En el caso de redes a distancia, es necesario utilizar un router en cada red. (Ver Figura 55).

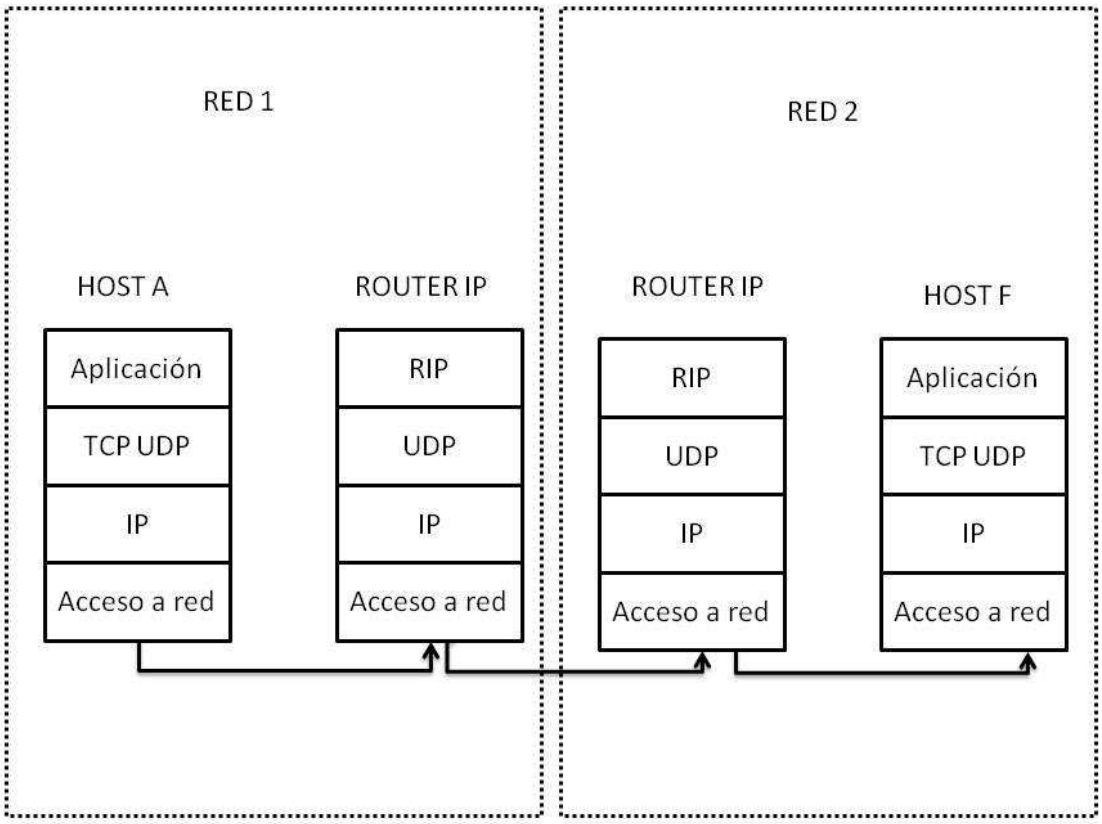


Figura 55

Los routers deben cumplir dos tareas principales:

a) Reenvío de paquetes (*forwarding*): cuando un paquete llega a un ruteador, este tiene que pasar el paquete a la salida apropiada, pero ya a nivel de puerto físico.
b) Encaminamiento de paquetes (*routing*): mediante el uso del protocolo RIP, tiene la capacidad de construir las tablas de información (dentro del router) que permiten que este pueda determinar la ruta que deben seguir los paquetes (es decir, por qué puerto físico se deben enviar los paquetes).

Como ya el lector puede vislumbrar, el protocolo IP tiene que estar dentro de cada *host* y en los enrutadores. En el *host*, básicamente, construyen los datagramas y asignan las direcciones IP; y en los routers, direccionan paquetes y construyen las tablas de enrutamiento.

En la figura 56, se plantea una situación simple para que se pueda comprender mejor el concepto de ruteo.

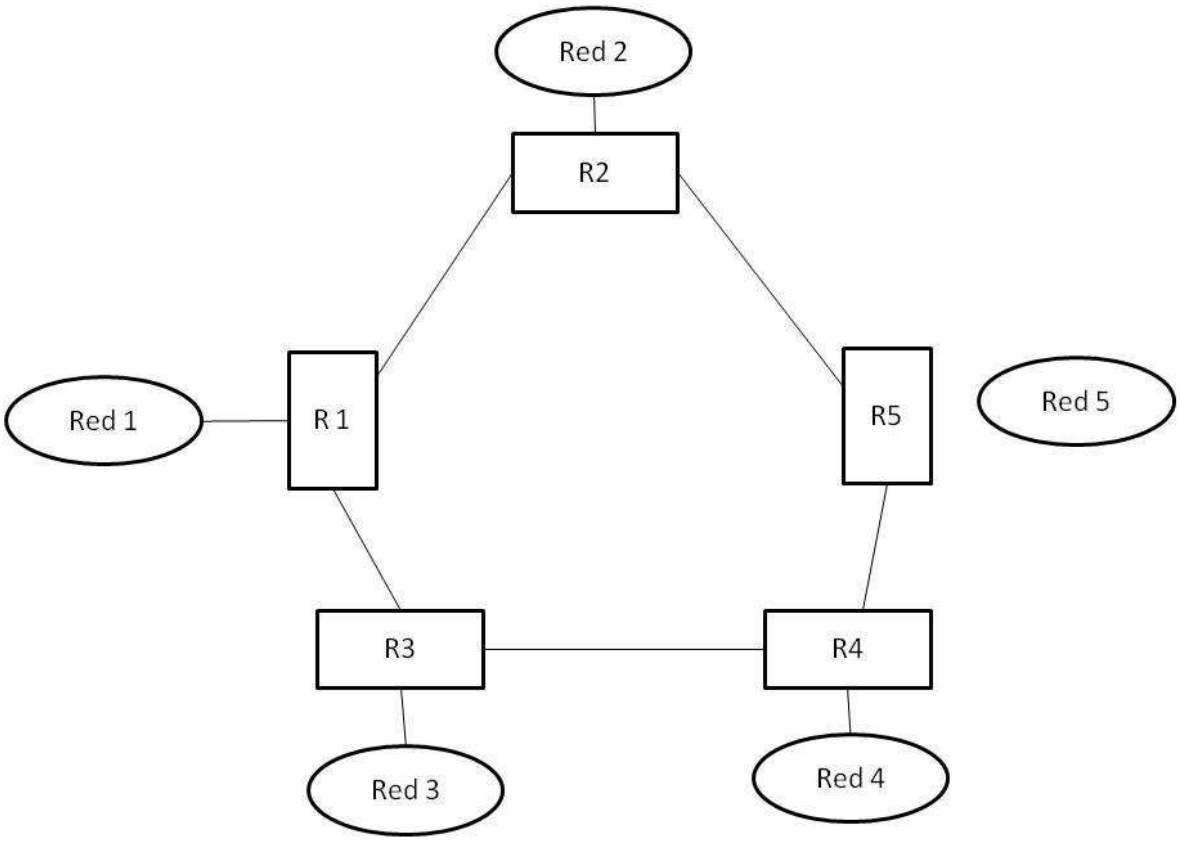


Figura 56

Si un *host* de la Red 1 le envía un datagrama a un *host* de la Red 5, el ruteador 1 debe tener la capacidad de discernir si el paquete lo saca por el puerto físico 1 con dirección al ruteador 2 o lo saca por el puerto físico 2 con dirección al ruteador 3. En este caso, se puede ver a simple vista que por el puerto 2 deberá efectuar un salto más que por el puerto 1.

Sin embargo, como ya se dijo anteriormente, el protocolo IP no transporta el paquete, esta tarea la realiza el protocolo de acceso a la red. Esto hace ver que entre el protocolo IP y los protocolos de acceso a red debe haber una fuerte relación. Por este motivo, existen importantes protocolos del nivel de acceso a red que no se llevan muy bien con el protocolo IP y deben buscarse soluciones de compromiso, como el caso de ATM.

7.4.1.1.5-RIP

El Protocolo de Información de Enrutamiento RIP (*Routing Information Protocol)* fue creado en 1970 por la empresa Xerox. Es un protocolo utilizado por los routers, aunque también pueden actuar en equipos, para intercambiar información acerca de redes IP. Es un protocolo de vector de distancias, ya que mide el número de «saltos»

como métrica hasta alcanzar la red de destino. El límite máximo de saltos en RIP es de 15, 16 se considera una ruta inalcanzable o no deseable.

Los fundamentos de RIP son los siguientes:

a) Cuando un router se inicializa, las únicas rutas que detecta son las de las redes a las que está directamente conectado.
b) El router difunde la formación de todas las redes a las que está directamente conectado (*triggered updates*).
c) Los routers escuchan la difusión de los *triggered updates*. Los routers se informan de las redes que no tienen directamente conectadas.
d) RIP construye las métricas para alcanzar a cada router en función de la cantidad de saltos.
e) Los *triggered updates* se envían cada 30 segundos.
f) RIP utiliza el protocolo UDP y el puerto 520 para enviar los mensajes (Ver Figura 55).
g) Las rutas se mantienen vivas durante 180 segundos, lo que corresponde a 6 intercambios de información.

Se realizará un simple ejemplo de cómo se difunde la información de cada router al resto de los routers de la red. Se utilizarán números simples en lugar de direcciones IP para simplificar el ejemplo:

En la figura 57, se muestra una red con cuatro nodos (A, B, C y D) con sus respectivas redes asociadas (1, 2, 3, 4, 5, 6, 7 y 8).

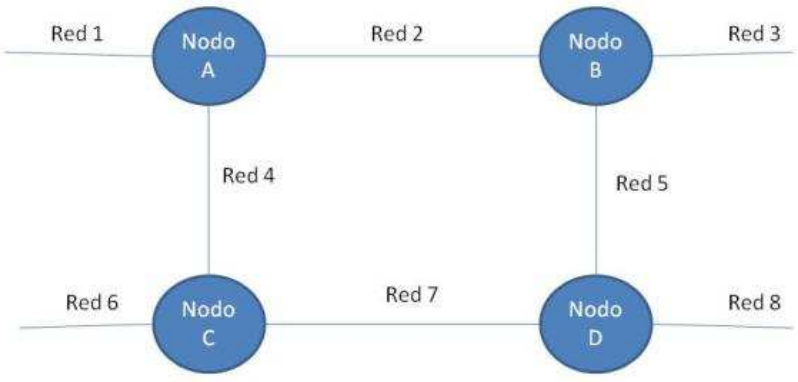


Figura 57

En la figura 58, se muestra el primer vector que construye cada nodo (router) en función de las redes que tiene directamente conectadas.

Nodo A		Nodo B		Nodo C		Nodo D	
Red	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**
1		2		4		5	
2		3		6		7	
4		5		7		8	

Figura 58

En la figura 59, se muestra la primer difusión que efectúa el nodo A a los nodos que tiene directamente conectados (B y C).

A les informa a B y a C

Nodo A		Nodo B		Nodo C		Nodo D	
Red	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**
1		2		4		5	
2		3		6		7	
4		5		7		8	
		1	1	1	1		
		4	1	2	1		

Figura 59

En la figura 60, se muestra la difusión que efectúa el nodo B a los nodos que tiene directamente conectados (A y D).

B les informa a A y a D

Nodo A		Nodo B		Nodo C		Nodo D	
Red	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**
1		2		4		5	
2		3		6		7	
4		5		7		8	
3	1	1	1	1	1	2	1
5	1	4	1	2	1	3	1
						1	2
						4	2

Figura 60

En la figura 61, se muestra la difusión que efectúa el nodo D a los nodos que tiene directamente conectados (B y C).

D les informa a B y a C

Nodo A		Nodo B		Nodo C		Nodo D	
Red	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**
1		2		4		5	
2		3		6		7	
4		5		7		8	
3	1	1	1	1	1	2	1
5	1	4	1	2	1	3	1
		7	1	5	1	1	2
		8	1	8	1	4	2
				3	2		

Figura 61

En la figura 62, se muestra la difusión que efectúa el nodo C a los nodos que tiene directamente conectados (D y A).

C les informa a D y a A

Nodo A		Nodo B		Nodo C		Nodo D	
Red	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**
1		2		4		5	
2		3		6		7	
4		5		7		8	
3	1	1	1	1	1	2	1
5	1	4	1	2	1	3	1
6	1	7	1	5	1	1	2
7	1	8	1	8	1	4	1
8	2			3	2	6	1

Figura 62

En la figura 63, se muestra la difusión que efectúa el nodo A a los nodos que tiene directamente conectados (B y C).

A les informa a B y a C

Nodo A		Nodo B		Nodo C		Nodo D	
Red	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**	**Red**	**Métrica**
1		2		4		5	
2		3		6		7	
4		5		7		8	
3	1	1	1	1	1	2	1
5	1	4	1	2	1	3	1
6	1	7	1	5	1	1	2
7	1	8	1	8	1	4	1
8	2	6	2	3	2	6	1

Figura 63

Las difusiones seguirán efectuándose, pero mientras no existan cambios en las relaciones entre los routers y las redes, no habrá alteraciones a los vectores que han quedado definidos para cada router.

Actualmente la versión del protocolo RIP más nueva es la RIPng (*next generation*) que entre otras cosas incorpora la compatibilidad con IPv6.

7.4.1.1.6-NAT

NAT (*Network Address Translation* - Traducción de Dirección de Red) es un mecanismo utilizado por el ruteador IP para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles. Este protocolo se desarrolló por la falta de direcciones IPv4. Fundamentalmente, se utiliza para unir una red privada a una red pública (Internet). El ruteador es el único equipo de la red privada que tiene asignada una dirección IP pública (cuyo responsable último es ICANN). Cuando la red privada envía a la red pública un datagrama, el ruteador convierte la dirección de origen (de la red interna) a la dirección IP pública del ruteador. El ruteador guarda información de la IP privada y el puerto (que se encuentra en la cabecera del encapsulado TCP) que originó el datagrama para que cuando llegue la respuesta pueda direccionarlo correctamente al dispositivo que le dio origen. Los paquetes que llegan al ruteador desde la red pública vienen con destino de la IP pública del ruteador y un determinado puerto de destino. El ruteador cambia la dirección IP pública del datagrama y la cambia por la IP destino de la red privada basándose en el puerto TCP o UDP.

Este mecanismo es similar a cuando se llama a un conmutador telefónico de una empresa. Para llamar al conmutador, se disca el número público de la red telefónica y luego la operadora pasa la comunicación al número de teléfono de la red interna.

7.4.1.1.7-DHCP

El protocolo de configuración dinámica de *host* DHCP (*Dynamic Host Configuration Protocol*) es un protocolo que complementa al protocolo IP, ya que permite a los *hosts* de una red IP obtener sus parámetros de configuración automáticamente. El router de la red tiene una funcionalidad llamada DHCP, la cual sirve para asignar dinámicamente números de IP a los *hosts* dentro de un rango parametrizable dentro del servicio. Cada vez que un *host* se conecta a la red, si este lo requiere, el servicio DHCP del router le asigna una dirección IP que durará en ese *host* hasta que este se desconecte. El servicio de DHCP contiene una función de trazabilidad de cada IP otorgada. Es decir que guarda una historia de quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

7.4.1.2-IPv6

Como ya se dijo, la principal necesidad de crear un nuevo protocolo para reemplazar IPv4 fue el agotamiento de la direcciones IPv4. Con la RFC 1752 de 1994, se aprobó el nuevo protocolo IPv6. IPv6 es absolutamente compatible con IPv4, por lo que pasar de IPv4 a IPv6 no implica la necesidad de una fecha determinada para su migración, esta puede ir haciéndose durante el transcurso del tiempo. Las direcciones IPv4 pueden ir incrustadas dentro de la dirección IPv6. Los ruteadores deben tener la capacidad de manejar ambos protocolos hasta que se decida discontinuar el uso del IPv4.

7.4.1.2.1-Formato de IPv6

Ya se ha dicho que IPv6 permite generar 340.282.366.920.938.463.463.374.607.431.768.211.456 (2^{128} o 340 sestillones) de direcciones IP. Las direcciones están formadas por 128 bits. Se prevén tres tipos de direcciones: *unicast*, *anycast* y *multicast*.

Unicast: Es un identificador para una única interfaz. Un paquete con una dirección unicast se entregará únicamente a la dirección indicada. Es la equivalente a la actual dirección IPv4.

Anycast: Es un identificador de un conjunto de interfaces. Un paquete enviado a esta dirección es entregado a la dirección más próxima. También puede crearse un esquema de redundancia para entregar en otra de las interfaces de igual dirección IP bajo un esquema de ruteo, básicamente, como medida de seguridad ante la salida de servicio de alguna interfaz.

Multicast: Es una dirección IP que identifica a un grupo de interfaces. Todas las interfaces con la misma dirección IP reciben los mismos paquetes. Desaparece la dirección broadcast del esquema IPv4

La dirección IP se representa por 8 valores de 16 bits cada uno (x.x.x.x.x.x.x.x). Este espacio de 16 bits permite 65.535 valores que en IPv6 se expresa en hexadecimal (FFFF). La dirección IPv6 se ve como 8 valores separados por un punto de cuatro dígitos hexadecimales, por ejemplo: 2001:0db8:0000:0000:0000:0000:1428:57ab

Existen una serie de reglas que permiten reducir el número de dígitos a ingresar cuando se establece una dirección IP, pero que son solo a efectos de practicidad.

IPv6 permite la gestión de máscara igual que como funciona en IPv4.

Dirección unicast global

Estas son las direcciones utilizadas para la comunicación global y su formato es el siguiente (Ver Figura 64):

3 bits	13 bits	8 bits	24 bits	16 bits	64 bits
011	TLA	RES	NLA	SLA	INTERFASE ID

Figura 64

El 011 indica que es una dirección unicast global.
El *Top Level Agregato*r (TLA) son direcciones asignadas por los Registros Regionales (RIRs), que dependen de ICANN, a organizaciones. Luego, los TLA asignan direcciones a los NLA.

El RES es una reserva de bits para futura necesidad de ampliación de los TLA.

Los *Next Level Agregator* (NLA) son direcciones de 24 bits donde las organizaciones (generalmente proveedores de servicios de Internet ISPs) pueden crear sus propios niveles jerárquicos. Los NLA asignan los SLA.

Los *Site Local Agregator* (SLA) son direcciones asignadas por los NLA a sus clientes para que estos creen sus propias redes para lo cual cuentan con 16 bits.

Los TLA y los NLA pueden usar SLA para sus propias redes.

Los últimos 64 bits se asemejan a los actuales *hosts* de la IPv4. La diferencia es que no se utilizan números arbitrarios, sino que se estipulan dos formas de asignación:

- Se le asigna el mismo número del hardware de la interfaz (MAC).
- Puede ser un identificador basado en IEEE EUI-64 asignado por un DHCPV6.

Dirección no especificada

Es la dirección 0.0.0.0.0.0.0.0 que se asigna a un equipo no incorporado a la red.

Dirección *loopback*

Es la dirección 0.0.0.0.0.0.0.1 que equivale a la dirección 127.0.0.1 de IPv4. Tiene muchos usos como autoreferencia, el más común es cuando, por ejemplo, en una máquina se instala un servicio WEB o un servicio FTP la dirección 0.0.0.0.0.0.0.1 apunta a la misma máquina para buscar el servicio.

Dirección base-IPv4

Es cuando se incrusta una dirección IPv4 dentro de una dirección IPv6, por ejemplo, 0.0.0.0.192.1.4.1 que equivale a la 192.1.4.1 de la IPv4.

Dirección unicast local

Son direcciones que se asemejan a las IPv4 de asignación privada (no públicas). Existen dos tipos de direcciones unicast locales: la Local de Enlace (*Link-local*) y la Local de Sitio (*Site-Local*).

Dirección Local de Enlace

Una dirección de enlace local es una dirección IP creada únicamente para comunicaciones dentro de una red local. Se identifican por sus primero 10 bits que deben ser igual a 1111111010. Los routers (que unen redes) no rutean los paquetes con estas direcciones. Estas direcciones se autoasignan, es decir que la misma computadora se genera su propia dirección (en este caso solo se asignan los valores de la dirección IP y la máscara de red) o bien se utiliza un servicio DHCPv6 mucho más completo que le permite al host identifica puerto de enlace, servicios DNS, etcétera (Ver Figura 65). Antes de asignarse la dirección, el host chequea que no exista ningún vecino dentro de la misma red con igual dirección IP por intermedio de un mensaje de tipo broadcast.

10 bits	54 bits	64 bits
1111111010	0s	Dirección de interfaz (MAC)

Figura 65

Direcciones Local de Sitio

Estas direcciones comienzan con 10 bits que las identifican (1111111011), no deben salir nunca a la red pública, son direcciones del sitio y manejan, además de la dirección de interfaz, un nivel superior de 16 bit para asignar número de red (Ver Figura 66). Se asignan con el servicio de DHCPv6 con IEEE EUI-64.

10 bits	38 bits	16 bits	64 bits
1111111011	0s	ID de subred	Dirección de interfaz (MAC)

Figura 66

7.4.1.2.2-El datagrama

La cabecera del datagrama IPv6 (Ver Figura 67) es mucho más simple que la cabecera de IPv4. Tiene una longitud fija de cuarenta octetos, lo que hace mucho más simple su tratamiento para los ruteadores y conmutadores. Sus campos son:

Figura 67

a) Número de versión: para este caso es 6.
b) Clase de tráfico: es similar al Tipo de Servicio en IPv4.
c) Etiqueta de flujo: es un campo destinado a brindar calidad de servicio para aquellas funcionalidades que requieren un flujo continuo de datos.
d) Longitud del paquete: aquí se informa la longitud del paquete que es variable y no puede exceder los 65.536 bytes.
e) Siguiente cabecera (8 bits), también llamada cabecera de extensión: el uso de un formato flexible de cabeceras de extensión opcionales es una idea innovadora que permite ir añadiendo funcionalidades de forma paulatina. Este diseño aporta gran eficacia y flexibilidad, ya que se pueden ir definiendo funcionalidades en el tiempo sin necesidad de alterar la estructura de la cabecera principal, una tarea que acarrearía muchas dificultades. Estas extensiones se colocan entre la cabecera fija o principal y la carga útil. Para esta funcionalidad, este campo indica la longitud de la próxima cabecera (Ver Figura 68). Hasta el momento existen 8 cabeceras de extensión.
f) Límite de saltos: es similar a Tiempo de Vida de la cabecera IPv4.

Figura 68

Hasta el momento, existen 8 cabeceras de extensión definidas (Ver Figura 69).

Cabecera de Extensión	Descripción	RFC
Opciones salto a salto (*Hop-By-Hop Options*)	Contiene datos que deben ser examinados por cada nodo a través de la ruta de envío de un paquete.	RFC 2460
Enrutamiento (*Routing*)	Métodos para especificar la forma de rutear un datagrama (Usado con IPv6 móvil).	RFC 2460, RFC 6275, RFC 5095
Cabecera de fragmentación (*Fragment*)	Contiene parámetros para la fragmentación de los datagramas.	RFC 2460
Cabecera de autenticación (*Authentication Header (AH)*)	Contiene información para verificar la autenticación de la mayor parte de los datos del paquete (Ver IPsec).	RFC 4302
Encapsulado de seguridad de la carga útil (*Encapsulating Security Payload (ESP)*)	Lleva la información cifrada para una comunicación segura (Ver IPsec).	RFC 4303
Opciones para el destino (*Destination Options*)	Información que necesita ser examinada solamente por los nodos de destino del paquete.	RFC 2460
No Next Header	Indica que no hay más cabeceras.	RFC 2460

Figura 69

7.5-IEEE 802.3

El primer antecedente de una red IEEE 802.3 fue la red ALOHA creada en la Universidad de Hawái, en 1970. Al igual que el grupo ARPANET, la red ALOHA se construyó con fondos de DARPA.

ALOHA usaba una transmisión de radio, esto se debía a que los diferentes centros de investigación estaban repartidos en varias islas, por lo que se buscaba un sistema de transmisión de datos inalámbrico. La característica más importante de esta red fue que usaban un medio compartido para la transmisión. ALOHA usaba para todos los nodos la misma frecuencia. Esto suponía la necesidad de algún tipo de sistema para controlar quién podían emitir y en qué momento. Es como imaginar un puente por donde pasa un solo vehículo, alguien debería coordinar la utilización del medio común. El esquema de ALOHA era muy simple: cuando dos estaciones trataban de emitir al mismo tiempo, ambas transmisiones colisionaban, y los datos tenían que ser reenviados nuevamente. ALOHA demostró que es posible tener una red útil sin necesidad de una intermediación.

Los dos principios de la red ALOHA fueron:

- Si tienes datos que enviar, envíalos.
- Si el mensaje colisiona con otra transmisión, intenta reenviarlos más tarde.

Desde luego que pensar en una solución así para un puente donde pasa un solo vehículo sería de resultados catastróficos, pero este mecanismo utilizado en las comunicaciones resultó ser muy eficiente. El esquema de ALOHA puro tenía un rendimiento de un 18,4% de efectividad, pero, como veremos, esquemas mejorados de esta idea brindaron mucha más efectividad.

El concepto de ALOHA fue tomado, en el año 1980, por la empresa Xerox, que desarrolló esta idea bajo el nombre de Ethernet. Luego se unió a Digital e Intel y juntas desarrollaron la especificación Ethernet en un conjunto de normas que denominaron Ethernet Blue Book 1. Después emitieron una nueva especificación denominada Ethernet Blue Book 2, que más tarde, en 1983, fueron desarrolladas como normas estándar por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, *The Institute of Electrical and Electronics Engineers*). Esta norma fue llamada IEEE 802.3 de Acceso Múltiple con Detección de Portadora y Detección de Colisiones (CSMA/CD, *Carrier Sense*; *Multiple Access*; *Collision Detect*). Igualmente, este tipo de redes siguieron siendo denominadas como Ehternet.

El estándar IEEE 802.3 define la gama de tipos de cables que pueden ser usados en la red. Incluye pares de cables trenzado, cable coaxial, fibra óptica y circuitos impresos.

Además, especifica el tipo de señal y la velocidad. Con el correr de los años, se han ido desarrollando distintas variantes de la norma IEEE 802.3 para adaptarse a las nuevas tecnologías, agregándole funcionalidades y mejorando su rendimiento. En la figura 70, se desarrolla una lista de las distintas versiones de la norma IEEE 802.3…

Estándar IEEE 802.3	**Fecha**	**Descripción**
802.3	1983	10BASE5 10 Mbit/s sobre coaxial grueso (*thicknet*). Longitud máxima del segmento 500 metros - Igual que DIX, salvo que el campo de Tipo se sustituye por la longitud.
802.3a	1985	10BASE2 10 Mbit/s sobre coaxial fino (*thinnet* o *cheapernet*). Longitud máxima del segundo.
802.3b	1985	10BROAD36.
802.3c	1985	Especificación de repetidores de 10 Mbit/s.
802.3d	1987	FOIRL (*Fiber-Optic Inter-Repeater Link*), enlace de fibra óptica entre repetidores.
802.3e	1987	1BASE5 o StarLAN.
802.3i	1990	10BASE-T 10 Mbit/s sobre par trenzado no blindado (UTP). Longitud máxima del segmento 150 metros.
802.3j	1993	10BASE-F 10 Mbit/s sobre fibra óptica. Longitud máxima del segmento 1000 metros.
802.3u	1995	100BASE-TX, 100BASE-T4, 100BASE-FX Fast Ethernet a 100 Mbit/s con autonegociación de velocidad.
802.3x	1997	Full Duplex (Transmisión y recepción simultáneas) y control de flujo.
802.3y	1998	100BASE-T2 100 Mbit/s sobre par trenzado no blindado (UTP). Longitud máxima del segmento 100 metros.
802.3z	1998	1000BASE-X Ethernet de 1 Gbit/s sobre fibra óptica.
802.3ab	1999	1000BASE-T Ethernet de 1 Gbit/s sobre par trenzado no blindado.
802.3ac	1998	Extensión de la trama máxima a 1522 bytes (para permitir las "Q-tag"). Las Q-tag incluyen información para 802.1Q VLAN y manejan prioridades según el estándar 802.1p.
802.3ad	2000	Agregación de enlaces paralelos. Movido a 802.1AX.
802.3ae	2003	Ethernet a 10 Gbit/s ; 10GBASE-SR, 10GBASE-LR.
802.3af	2003	Alimentación sobre Ethernet (PoE).
802.3ah	2004	Ethernet en la última milla.
802.3ak	2004	10GBASE-CX4 Ethernet a 10 Gbit/s sobre cable biaxial.
802.3an	2006	10GBASE-T Ethernet a 10 Gbit/s sobre par trenzado no blindado (UTP).
802.3aq	2006	10GBASE-LRM Ethernet a 10 Gbit/s sobre fibra óptica multimodo.

802.3ap	2007	Ethernet de 1 y 10 Gbit/s sobre circuito impreso.
802.3ar	2008	Gestión de Congestión.

Figura 70

En forma genérica, las reglas del protocolo IEEE 302.3… son:

- Si el medio de transmisión está libre, se puede transmitir.
- Si el medio está ocupado, entonces se espera hasta que se libere para poder transmitir.
- Si se encuentra el medio de transmisión libre, se intenta la transmisión. Si al transmitir se produce una colisión, enseguida se detiene la transmisión, y entonces se envía una señal de JAM a todos los miembros de la red, después de la cual todos detienen la transmisión completamente por un tiempo seleccionado aleatoriamente para cada nodo de la red.
- Si al volver a tratar de transmitir ocurriera de nuevo una colisión, entonces se recurre al retardo exponencial binario, el cual consiste en duplicar el tiempo de espera inicial. Y si aun así se repitiera la colisión, de nuevo se duplica el tiempo anterior, y así sucesivamente hasta que se pueda transmitir la trama o se llegue al límite de reintentos parametrizados en la red después del cual la trama se descarta.

Este protocolo se implementa por intermedio de unos circuitos impresos denominados Tarjetas de Interfaz de Red (NIC, del inglés *network interfaz card*). Cada NIC tiene un identificador único denominado dirección MAC (*Media access control*). Esta dirección está formada por 48 bits (seis octetos). Los primeros tres octetos identifican al fabricante de la NIC y se lo llama identificador de organización único (OUI, *Organizationally Unique Identifier*). Los otros tres octetos son otorgados por la organización que fabrica la NIC.

El método de codificación binaria es Manchester, que es por desplazamiento de frecuencia (Ver Punto 4.2.2).

El formato de la trama es el siguiente:

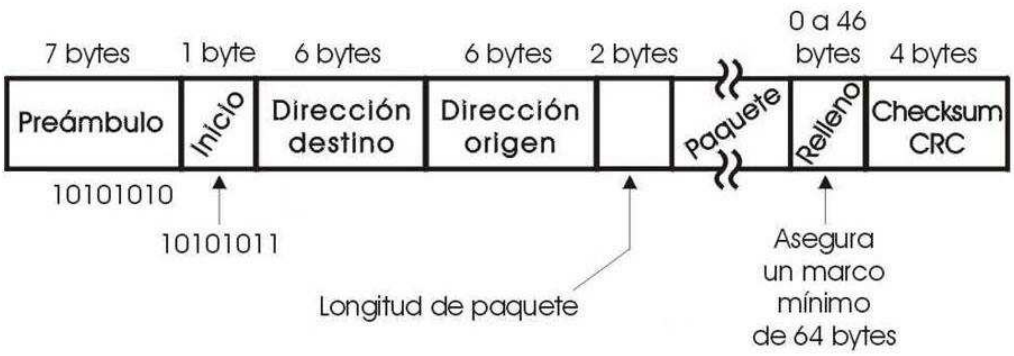


Figura 71

Preámbulo**:** Son siete bytes (48 bits) donde se alternan unos y cero con el objetivo de sincronizar el reloj de la transmisión (cada octeto contiene 10101010). Hay que tener en cuenta que la comunicación es serial y asíncrona (no se transmite una señal de reloj independiente), por lo tanto, para que el receptor pueda identificar los bits que debe leer, primero debe identificar la velocidad, la amplitud de la señal y las fases. Para resolver este problema, el transmisor le envía una ráfaga de 10 unos y ceros. Con ellos el receptor intentará ajustar su propio reloj a la frecuencia y la fase del trasmisor.

Start Frame Delimiter (SFD) o Inicio**:** Es un solo octeto (10101011) que indica el comienzo de la trama. Es poco probable que el reloj del receptor se sincronice inmediatamente, seguramente perderá algunos bits del preámbulo para lograrlo. Cuando esto ocurre, no puede identificar cuántos bits perdió y, por ende, tampoco sabe cuándo termina el preámbulo y comienzan los datos. Para eso se agrega otro byte similar a los del preámbulo, solo que termina con “11” en lugar de los “10” habituales.

Destino**:** Este dato es de seis bytes. Aquí se informa la dirección física de destino (la dirección MAC).

Origen**:** Este dato es de seis bytes. Aquí se informa la dirección física del origen. Es decir la MAC de la NIC que está enviando los datos.

Como se puede identificar, no es suficiente con que el sistema conozca la dirección lógica del equipo de destino, también debe conocer la dirección hardware (MAC) del destinatario. La solución a este problema la provee el protocolo ARP.

Longitud del campo de Datos/Tipo**:** Se utilizan dos bytes para identificar el largo del paquete de datos. Esto es necesario ya que la trama no tiene una identificación de fin de datos.

Paquete**:** Es la información que se está transportando, tiene una longitud máxima de 1500 bytes.

Relleno**:** Son 46 bytes que se reservan para considerar en la trama si es que esta no llega a tener una longitud mínima de 64 bytes. Esta longitud mínima es necesaria para que las NICs de la red puedan detectar las colisiones.

Checksum CRC**:** significa suma de chequeo (se calcula por control de redundancia cíclica CRC). Es realizada por el emisor y la informa en este campo de cuatro bytes, luego el receptor vuelve a realizar la suma y la compara con el valor recibido en este campo; si son iguales, significa que la trama ha llegado completa.

7.5.1-Topologías IEEE 802.3

En esencia la topología de IEEE 802.3 es de bus, pero la forma de implementarla, en la mayoría de los casos, asemeja a una topología estrella.

Cuando se utiliza un cable coaxial, la topología se puede describir, inconfundiblemente, como de bus (Ver Figura 36). En esta forma, los dispositivos son conectados a un único tramo de cable. Este cable es el medio común para todos los dispositivos conectados y transporta todas las transmisiones entre los dispositivos.

El problema más común de esta configuración es que una falla en cualquier parte del cable coaxial va a interrumpir la comunicación de todo el sistema.

Cuando se utilizan pares trenzados o de fibra óptica, los dispositivos son dispuestos en una topología estrella (Ver Figura 34); aunque, como se dijo anteriormente, esencialmente es una topología bus. En esta topología, los dispositivos individuales son conectados a un concentrador (*hub* o *switch*) central, formando un segmento. Las tramas de cada dispositivo son enviadas al concentrador y, dentro de él, se difunden a los otros dispositivos conectados. Este diseño permite operar lógicamente como un bus, pero físicamente el bus solo existe en el concentrador.

Esta forma de conectar los dispositivos simplifica la administración de la red, ya que cada tramo de cable conecta solo una NIC al concentrador. Si una NIC no puede comunicarse exitosamente con simplemente conectarla a otra salida del concentrador, es posible identificar si el problema es del dispositivo o del concentrador.

7.5.1-*Hub*

Ya se explicó en el punto anterior que una topología de estrella en el protocolo IEEE 802.3 requiere de un concentrador, uno de los posibles concentradores es el *hub* (Ver Figura 72). Es importante aclarar que este dispositivo es prácticamente obsoleto, pero es bueno comentarlo para el entendimiento del protocolo IEEE 802.3. Este es un dispositivo que se encarga de conectar entre sí todos los equipos de una red. El *hub* no es más que una toma de múltiple bocas (puertos) para conectores RJ45 que amplifica la señal de la red. Un *hub* funciona repitiendo cada trama de datos en cada uno de los puertos con los que cuenta, excepto en el que se recibe el paquete; de forma que todas las bocas tienen acceso a todos los datos que pasan por un *hub*. También, se encarga de enviar una señal de choque a todos los puertos si se detecta una colisión.

Figura 72

Para conectar cada placa de red (NIC) con el puerto del *hub*, se utilizan cables UTP sin blindaje (Ver Figuras 73 y 74).

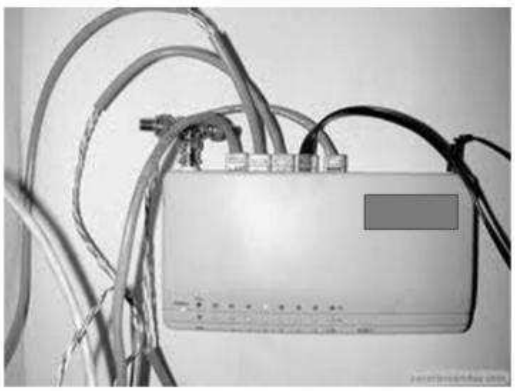


Figura 73

Figura 74

Si la red crece de forma tal que el *hub* que se ha instalado no cuenta con puertos suficientes, se pueden interconectar otros *hubs*. Al conectar otro concentrador a la red, hay que asegurarse de que el nuevo equipo sea de una velocidad compatible (Ver Figura 75).

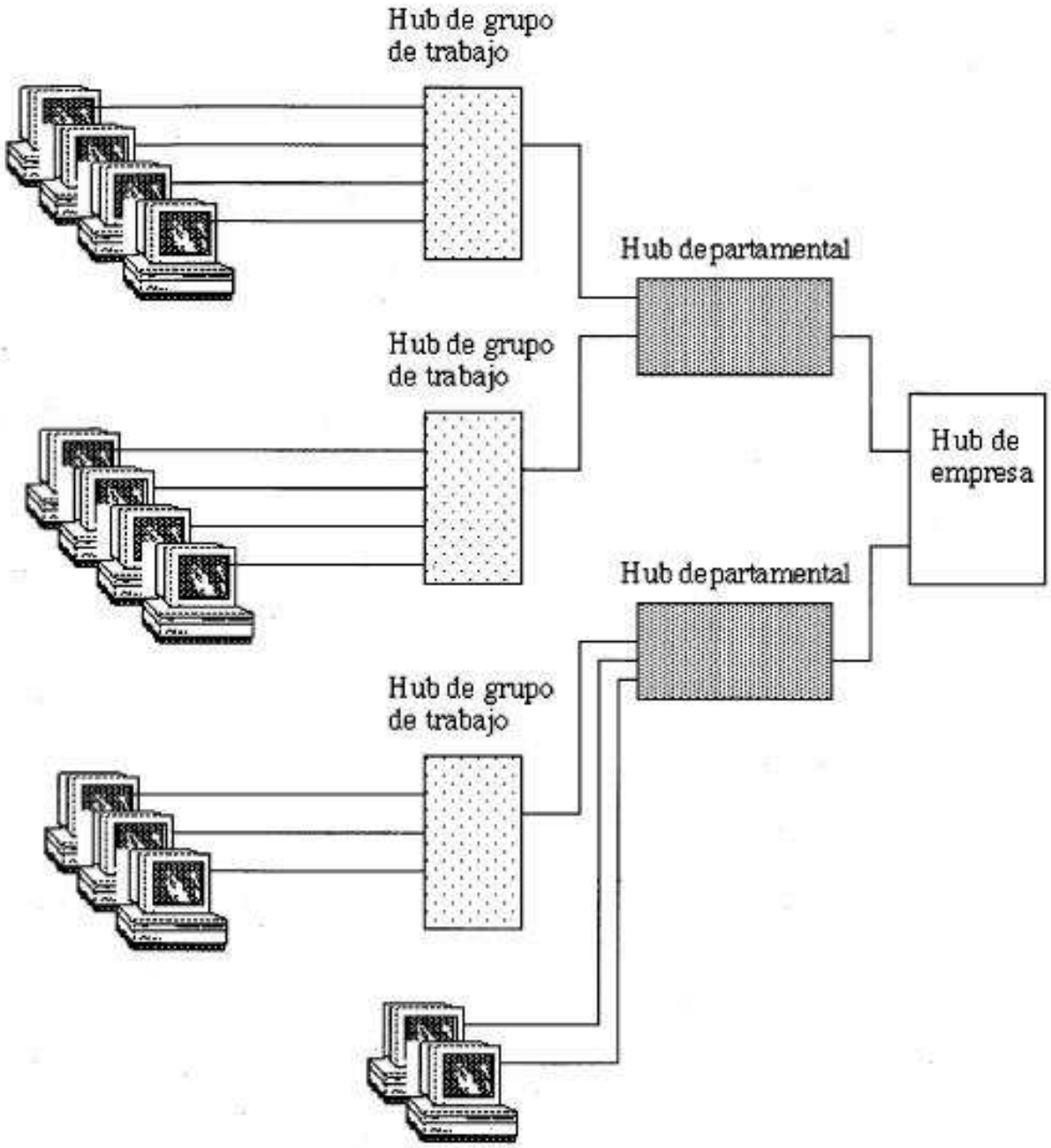


Figura 75

Por la baja performance de los *hubs*, debido a la gran cantidad de colisiones que dentro de él se producen, se han reemplazando por el *switch* que se asemeja mucho a los *hubs*, pero que están pensados para evitar las colisiones de las tramas.

7.5.2-*Switch*

Un *switch* es un dispositivo electrónico de interconexión de redes de computadoras que distribuye los datos a cada máquina de destino, a diferencia con el hub que envía todos los datos a todas las máquinas.

Concebido para trabajar en redes con una cantidad de máquinas ligeramente más elevado que el *hub*, el *switch* reduce sustancialmente las eventuales colisiones de paquetes. En este dispositivo, las tramas son enviadas desde la NIC de origen a la NIC de la computadora de destino. Cuando una trama entra por un puerto del *switch*, a diferencia del *hub* que las comunica a todos los demás puertos, es dirigida directamente al puerto de destino y los demás puertos no intervienen en la transmisión, quedando liberados para atender sus propias comunicaciones. El problema de colisión se reduce a cuando dos NICs envían tramas a una misma NIC. En el puerto que conecta la NIC de salida, se produce la colisión.

El *switch* es prácticamente igual que el *hub* en cuanto al tipo de conectores, cables y formas de conexión que utiliza.

7.5.3-IEEE 802.11 (*Wireless LAN*)

La variante inalámbrica del protocolo Ethernet es el estándar IEEE 802.11, también llamada WIFI. Esta tecnología es muy similar a IEEE 802.3 en muchos aspectos. El método de acceso al medio de IEEE 802.11, a diferencia con IEEE 802.3, es MACAW. El problema de las líneas inalámbricas es que las colisiones se producen en el aire y, por lo tanto, no pueden ser escuchadas. Las transmisiones se realizan mediante señales de radio frecuencias. Pero, a diferencia de una red en forma de bus donde el medio es absolutamente compartido, las redes inalámbricas son más parecidas a una red en forma de estrella, donde existe un nodo coordinador, cuya figura está representada por el Wireless Access Point (WAP).

MACAW (*Medium Access Collision Avoidance for Wireless*) consiste en un método de solicitud de permisos y emisión de autorizaciones. Cualquier estación antes de transmitir una trama de datos (esta relación es siempre entre la estación de trabajo y el WAP) envía una trama pidiendo autorización al nodo receptor para hacerla. El emisor emite un mensaje «*Request To Send*», que tiene que ser contestado por el receptor con un mensaje «*Clear To Send*» para poder hacer la retransmisión. Cuando el emisor recibe el mensaje «*Clear To Send*», ya puede hacer la transmisión real con los datos. El resto de los dispositivos que desean trasmitir datos, al detectar los mensajes «*Request To Send»* o «*Clear To Send*» saben que el canal está ocupado. Cuando una trama de datos acaba de ser transmitida, el receptor envía un mensaje de acuse de recibo «*ACK*» al emisor; de esta forma, el resto de los equipos ya saben que pueden volver a intentar su transmisión. Cuando el canal está libre, varias estaciones pueden intentar enviar un «*Request To Send*» simultáneamente y colisionar en el aire, por lo cual ninguna de las ellas recibirá un «*Clear To Send*».

Si se envía un «*Request To Send*» y el canal está ocupado, se espera un espacio de tiempo, y así sucesivamente, pero reduciendo el espacio de tiempo de espera. De esta forma, se garantiza una cantidad mínima de colisiones.

Existe una gran variedad de versiones de este protocolo que se describen a continuación:

Nombre del estándar	Descripción
802.11a	El estándar 802.11 (llamado WiFi 5) admite un ancho de banda superior (el rendimiento total máximo es de 54 Mbps, aunque en la práctica es de 30 Mbps). El estándar 802.11a provee ocho canales de radio en la banda de frecuencia de 5 GHz.
802.11b	El estándar 802.11 ofrece un rendimiento total máximo de 11 Mpbs (6 Mpbs en la práctica) y tiene un alcance de hasta 300 metros en un espacio abierto. Utiliza el rango de frecuencia de 2,4 GHz con tres canales de radio disponibles.
802.11c	El estándar combinado 802.11c no ofrece ningún interés para el público general. Es solamente una versión modificada del estándar 802.1d que permite combinar el 802.1d con dispositivos compatibles 802.11 (en el nivel de enlace de datos).
802.11d	El estándar 802.11d es un complemento del estándar 802.11 que está pensado para permitir el uso internacional de las redes 802.11 locales. Permite que distintos dispositivos intercambien información en rangos de frecuencia según lo que se permite en el país de origen del dispositivo.
802.11e	El estándar 802.11e está destinado a mejorar la calidad del servicio en el nivel de la *capa de enlace de datos*. El objetivo del estándar es definir los requisitos de diferentes paquetes en cuanto al ancho de banda y al retardo de transmisión para permitir mejores transmisiones de audio y video.

802.11f	El 802.11f es una recomendación para proveedores de puntos de acceso que permite que los productos sean más compatibles. Utiliza el *protocolo IAPP* que le permite a un usuario itinerante cambiarse claramente de un punto de acceso a otro mientras está en movimiento, sin importar qué marcas de puntos de acceso se usan en la infraestructura de la red. También se conoce a esta propiedad simplemente como *itinerancia.*
802.11g	El estándar 802.11g ofrece un ancho de banda elevado (con un rendimiento total máximo de 54 Mbps, pero de 30 Mpbs en la práctica), en el rango de frecuencia de 2,4 GHz. El estándar 802.11g es compatible con el estándar anterior, el 802.11b, lo que significa que los dispositivos que admiten el estándar 802.11g también pueden funcionar con el 802.11b.
802.11h	El estándar 802.11h tiene por objeto unir el estándar 802.11 con el estándar europeo (HiperLAN 2, de ahí la *h* de 802.11h) y cumplir con las regulaciones europeas relacionadas con el uso de las frecuencias y el rendimiento energético.
802.11i	El estándar 802.11i está destinado a mejorar la seguridad en la transferencia de datos (al administrar y distribuir claves, y al implementar el cifrado y la autenticación). Este estándar se basa en el *AES* (estándar de cifrado avanzado) y puede cifrar transmisiones que se ejecutan en las tecnologías 802.11a, 802.11b y 802.11g.
802.11Ir	El estándar 802.11r se elaboró para que pueda usar señales infrarrojas. Este estándar se ha vuelto tecnológicamente obsoleto.
802.11j	El estándar 802.11j es para la regulación japonesa, lo que el 802.11h es para la regulación europea.

Figura 76

El formato genérico de la trama del protocolo 802.11 es el siguiente:

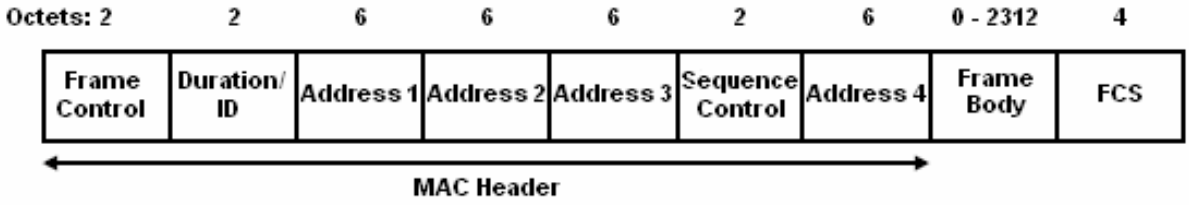


Figura 77

Lo interesante para destacar de esta trama es que tiene cuatro direcciones de las cuales solo se usan tres. Una es para identificar el BSSID (*Basic Service Set Identifier*), que es un identificador único de todos los paquetes de una WIFI y que se forma con la dirección MAC del WAP (*Wireless Access Point*) o punto de acceso a la red; otra se utiliza para la MAC de destino; y la última se utiliza para la MAC de origen. Los datos se encapsulan en el Frame Body.

7.5.3.1-*Wireless Access Point* (WAP)

Los equipos de la red WLAN (*Wireless LAN*) se conectan entre sí utilizando algunos de los protocolos anteriormente descriptos. El alcance de la conexión de RF puede variar entre 10 y 100 metros. Cada dispositivo inalámbrico se conecta a un equipo denominado WAP (Ver Figura 78). Un WAP posee su propio nombre. Este nombre es el SSID de la red (muchas redes WIFI pueden tener el mismo SSID). Normalmente, un WAP también puede conectarse a una red cableada y puede transmitir datos entre los dispositivos conectados a la red de cable y a los dispositivos inalámbricos.

Figura 78

El WAP puede estar unido a una red IEEE 802.3 conectándolo a un *switch* de la LAN, como se muestra en la figura 79.

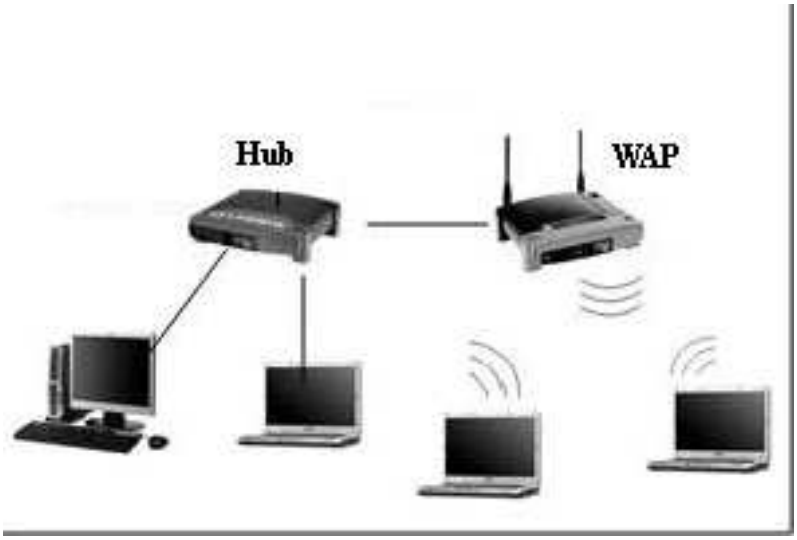


Figura 79

Muchas veces, los WAP se combinan dentro del mismo equipo con un *switch* o un router, como en la figura 80, al que normalmente se lo suele llamar router inalámbrico.

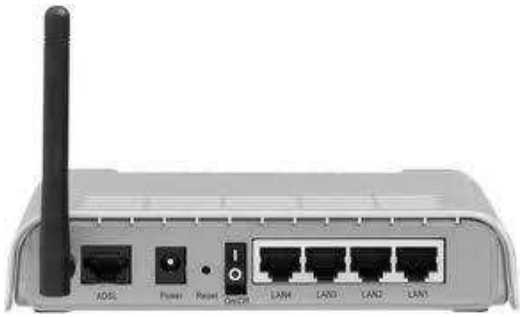


Figura 80

La figura anterior muestra un equipo con un punto de acceso WAP, cuatro bocas de RJ45 para protocolo IEEE 802.3 (como función de *switch*) y una boca para la conexión a Internet por la salida de ADSL.

7.5.4-ARP

Protocolo de Resolución de Direcciones, ARP (*Address Resolution Protoco*), es un protocolo definido en la RFC-826 y trabaja sobre redes IEEE 803.2... e IEEE 803.11…

Es responsable de encontrar la dirección hardware que corresponde a una determinada dirección IP siembre sobre su misma red.

La información se intercambia a nivel físico, y en este nivel, las direcciones IP no tienen valor, por este motivo, es necesaria la existencia de un protocolo que resuelva las direcciones IP y las convierta en direcciones físicas. Por ejemplo, un *host* que quiere enviar una trama del protocolo IEEE 802.3… a una determinada dirección IP, primero deberá convertir dicha dirección a una dirección MAC, que es la única que reconoce IEEE 802.3… Para tal fin, se envía un paquete (ARP *request*) en forma de *broadcast* (lo que significa que se envía a todos los *hosts* de la red que contiene la dirección IP por la que se pregunta, y se espera a que el *host* que tiene asignada dicha dirección IP responda (ARP *reply*) informando su respectiva dirección MAC. Cada *host* mantiene una tabla que relaciona la dirección IP con la dirección MAC. Esto es para evitar que cada vez que se intenta mandar una trama se tenga que realizar nuevamente un ARP *request* con la finalidad de reducir el retardo y la carga de la red. ARP permite a la dirección IP ser independiente de la dirección MAC. Cuando el *host* no logra localizar la dirección MAC, envía el paquete al router para que este pueda buscarla en otras redes.

El protocolo RARP realiza la operación inversa y se encuentra descripto en el RFC 903.

7.6-DOCSIS

DOCSIS es un protocolo para servicios de datos por cable (*Data Over Cable Service Interfaz Specification*). Fue desarrollado por CableLabs, con la contribución de las siguientes compañías: 3Com, ARRIS, BigBand Networks, Broadcom, Cisco, Conexant, Correlant, Harmonic, Intel, Motorola, Netgear, Technicolor, Terayon y Texas Instruments. Este grupo se denomina *Multimedia Cable Network System Partners Limited* (MCNS). Por ser una de las redes más importantes que conectan a los hogares y a muchas PyMEs con Internet, además de ser una de las que más futuro promete, se le dedicará un espacio destacado.

El cable que llega a los hogares para brindar servicios de televisión ahora se ha convertido en una de las redes de datos más importantes. Por este cable, se transfieren datos de alta velocidad para proveer, fundamentalmente, servicios de Internet (entre ellos, telefonía IP); pero no solo esto, sino que además se provee de servicios intensos, como video digital, TV de alta definición (HDTV) y video por demanda (VOD).

El protocolo que se encarga de toda esta tarea es DOCSIS, que se aplica sobre una infraestructura HFC que se refiere a las redes híbridas de fibra óptica y cable coaxial (*Hybrid Fiber Coaxial*). Este protocolo se ubica dentro del nivel de acceso a red

dentro de los protocolos del conjunto TCP/IP o en las capas 1 y 2 del estándar OSI. El 7 de agosto de 2006, salió la especificación DOCSIS 3.0, que soporte IPv6 y *channel bonding*, que permite utilizar varios canales simultáneamente, tanto de subida como de bajada, por lo que el nuevo protocolo llega a velocidades de descarga de datos de 160 Mbps y subidas a 120 Mbps.

DOCSIS se instala sobre una red de punto multipunto (Ver Punto 3.2), que exige una manejo muy distinto y en cierta forma más controlado que el protocolo Ehternet, que se instala sobre una red donde todos compiten por el medio

Una red de un prestador de servicios de cable módem tiene la siguiente topología (Ver Figura 81):

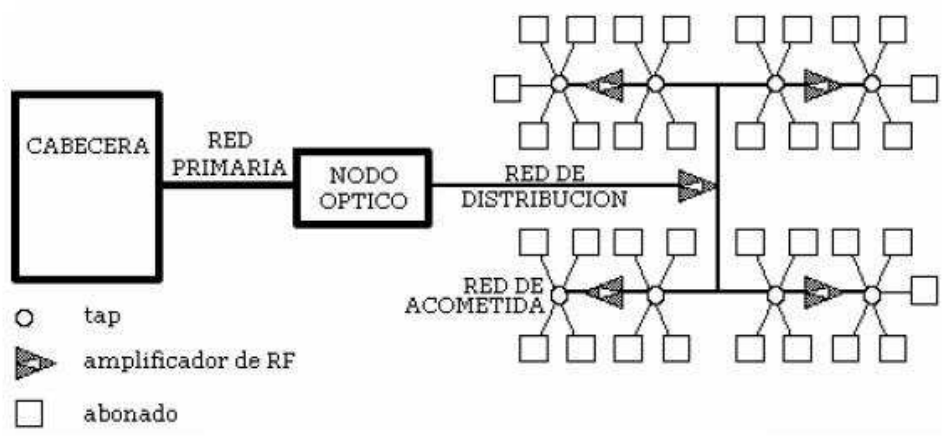


Figura 81

La cabecera es el CMTS o Sistema de Terminación de Cable Módem (*Cable Modem Termination System*) (Ver Figura 82), que envía las señales por la red troncal de fibra óptica. En el nodo óptico, se realiza la conversión de señal de óptica a eléctrica. Luego se inicia el proceso de distribución que tiene forma de árbol y termina en ramas que se dirigen a los hogares donde se encuentran los CM Cable Módem (Ver Figura 83). Los Tap son simples derivadores.

Esta estructura arbórea le permite al CMTS realizar el control y la administración de la red arbitrando el tránsito de la información sobre el canal.

El CMTS es el hardware instalado en el proveedor de servicios de televisión por cable que se conecta con los abonados al servicio. Estos equipos permiten conectar entre

4000 y 150.000 abonados. El CMTS asigna las direcciones IP, las puertas de enlace, los servicios DNS, etcétera.

El CMTS se conecta mediante el protocolo Ethernet hacia una red interna o bien se conecta con un enrutador IP para salir hacia otra red.

Figura 82

El CM es un tipo especial de módem diseñado para modular la señal de datos sobre una infraestructura de televisión por cable. Estos equipos son muy importantes en el funcionamiento de la red, ya que negocian con el CMTS la utilización del medio.

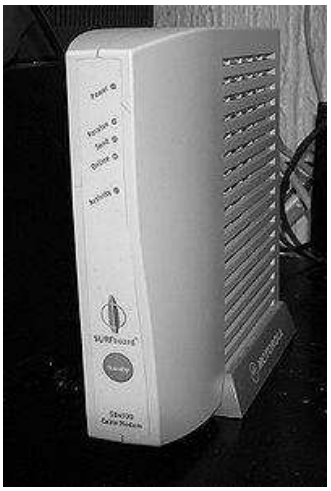


Figura 83

Las primeras versiones del protocolo DOCSIS manejaban solamente dos canales en forma independiente, uno corriente abajo (del CMTS hacia el CM), al que se denomina *Downstreem*; y otro corriente arriba (del CM hacia el CMTS), al que se denomina *Upstreem*. Los canales de bajada y subida son del mismo tamaño y tipo de

modulación, creando un sistema en banda simétrico. Desde la versión 3.0, DOCSIS puede manejar varios canales de bajada y varios canales de subida.

Downstreem

En el canal de bajada, DOCSIS trabaja dentro de los 88MHz y los 862MHz utilizando a 64-QAM y 256-QAM como técnicas de modulación.

El proceso de enviar información a los CM es relativamente simple, ya que no hay posibilidad de colisión en función de que los canales de comunicación están siendo administrados por el CMTS. DOCSIS requiere que los paquetes recibidos desde la red interna (Ethernet) o del ruteador sean transferidos en el canal de descenso, a través de la encapsulación en MPEG-TS (especificaciones ITU-T H.222.0) (Ver Figura 84). Una trama MPEG-TS tiene una longitud de 188 bytes, con los 4 primeros bytes como encabezado, reservando 184 bytes para la carga útil. Dentro del encabezado, existe un campo de 13 bits denominado PID, el cual identifica el tipo de información dentro de la carga útil. Cada CM solo selecciona tramas en función del valor del PID. Un paquete MPEG puede contener varios paquetes MACs (estos son los paquetes que construyen la capa física de DOCSIS) e incluso un paquete MAC puede estar distribuido entre varios paquetes MPEG. El paquete MAC de DOCSIS es de longitud variable. La longitud mínima es 72 bytes y el tamaño máximo es de 1500 bytes. Todos los CM reciben toda la información emitida por el CMTS, aunque siempre se los configura para recibir solamente los paquetes asignados a cada CM y rechazar el resto. Para direccionar los paquetes, utiliza el método de direccionamiento *multicast*. Cuando el CM se inicializa, el CMTS le asigna una dirección única a cada CM, pero, además, puede asignar valores adicionales de identificación a los CMs para formar grupos de direcciones. El CMTS puede en cualquier momento añadir o remover un módem de un grupo de direcciones. Usando este sistema, el CMTS puede construir varios grupos *multicast*, donde un CM asignado a un grupo recibirá los mismos paquetes de información que se envían al resto de los CMs del mismo grupo (esto facilita mucho el envío de los paquetes televisivos). Periódicamente el CMTS envía a los CMs un paquete de información que contiene los rangos de frecuencia que utilizará cada uno de los CM y demás parámetros operacionales.

La formación de la trama de bajada en MPEG-TS es de la siguiente forma:

A los datos cedidos por la capa de aplicación, se le agrega la cabecera de la capa de transporte; al paquete de la capa de transporte, se le agrega la cabecera de la capa de Internet y se forma lo que se denomina datagrama, que luego se entrega a la capa de acceso a red. Si se trata de un paquete que llega de la red interna del proveedor de servicios de cable (del *backend* del CMTS), además llegará con la cabecera y la cola del protocolo Ethernet. El CMTS toma este paquete y forma lo que se denomina paquete MAC. El paquete MAC tiene una cabecera que contiene el identificador del

módem de cable CM, que indica el contenido, la longitud del paquete y varios indicadores específicos.

El paquete MAC se envía a los CM encapsulados en tramas MPEG-TS. Estas tramas tienen una longitud fija de 204 bytes. Por lo tanto, un paquete MPEG-TS puede contener más de un paquete MAC o bien un paquete MAC puede estar distribuido entre varios paquetes MPEG.

La trama MPEG-TS se forma de la siguiente manera:

a) Se subdivide el paquete MAC en pedazos de 183 bytes, por lo que puede ocurrir que al último pedazo le quede un espacio vacío como en el ejemplo. Este espacio se completa con unos.
b) A cada pedazo de 183 bytes, se le agrega una cabecera MAC de 5 bytes que contiene información sobre el paquete y el destino.
c) Se le agregan 16 bytes en la cola con el FEC (*Forward Error Correction*), cuyo objetivo es permitir la corrección de errores para evitar la necesidad de que se reenvíen las tramas.

Las tramas MPEG-TS son enviadas continuamente a pesar de que no existan datos para ser enviados.

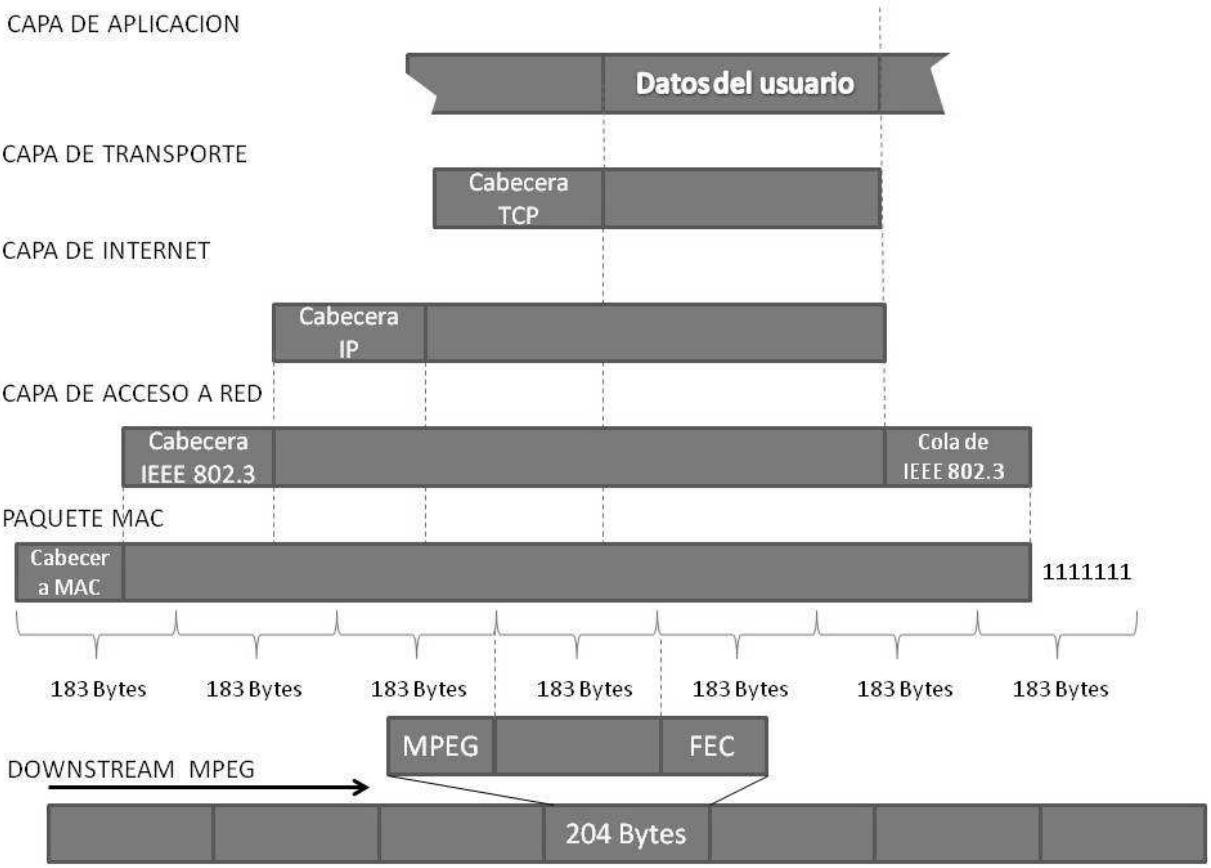


Figura 84

Upstream

Enviar información desde el CM hacia el CMTS es un proceso mucho más complejo que su inverso, ya que ahora la relación entre quienes envían y quien recibe es de muchos a uno. Esto obliga a que se utilice algún mecanismo de multiplexado del medio, de tal forma que se eviten, en lo posible, las colisiones. Este mecanismo es muy complejo, por lo que solamente se realizará una síntesis.

DOCSIS 3.0 trabaja, para subir información, en el rango de frecuencias de 5 MHz y 42MHz; dentro de este rango, abre múltiples canales de subida. Las técnicas de modulación son QPSK, 16-QAM y 64-QAM.

La transmisión desde el CM hacia el CMTS se realiza a través de una técnica de ranuración de los canales. A este método se lo denomina TDMA (Acceso múltiple por división de tiempo (del inglés, *Time Division Multiple Access*), muy similar al que realiza el protocolo ATM (Ver Figura 93). Cuando el CM se enciende, se establece una comunicación con el CMTS que se mantiene hasta que el CM se apague. En este proceso de conexión, se establecen los canales de subida y de bajada que serán utilizados, el mecanismo de seguridad a utilizar y la ranuración a utilizar. Estas ranuras son como contenedores de bits definidos por espacios de tiempo. Establecer el valor de esta ranura y la cantidad de ranuras que se le asignan a cada CM es uno de los procesos más complejos de este protocolo. En términos generales, lo que hace el CMTS es enviar periódicamente a cada CM un paquete que se denomina MAP, que le permite acomodar la red a los diferentes niveles de servicio que requiere cada abonado a la red QoS (*Quality of Service*). Cada vez que un CM pretende realizar una transmisión, debe requerir autorización al CMTS en función del último MAP recibido. El CMTS contesta con un nuevo MAP acorde a las necesidades del CM y de un proceso de optimización de la red.

El proceso de encapsulamiento es el siguiente (Ver Figura 85):

a) Se subdivide el paquete MAC en pedazos de mini-slots de tiempo (ranuras). Si algún mini-slot queda con espacio libre, se completa con unos.
b) A cada mini-slot se le agrega una cabecera de 16 bits que contienen números de secuencia.
c) Se le agregan 16 bytes en la cola con el FEC (*Forward Error Correction*), cuyo objetivo es permitir la corrección de errores para evitar la necesidad de que se reenvíen las tramas.
d) Por último, los mini-slots se concatenan con otros según lo establecido oportunamente por el MAP. La trasmisión se efectúa en forma de ráfagas

de mini-slots extremadamente calculadas por el CMTS para lograr la optimización del sistema.

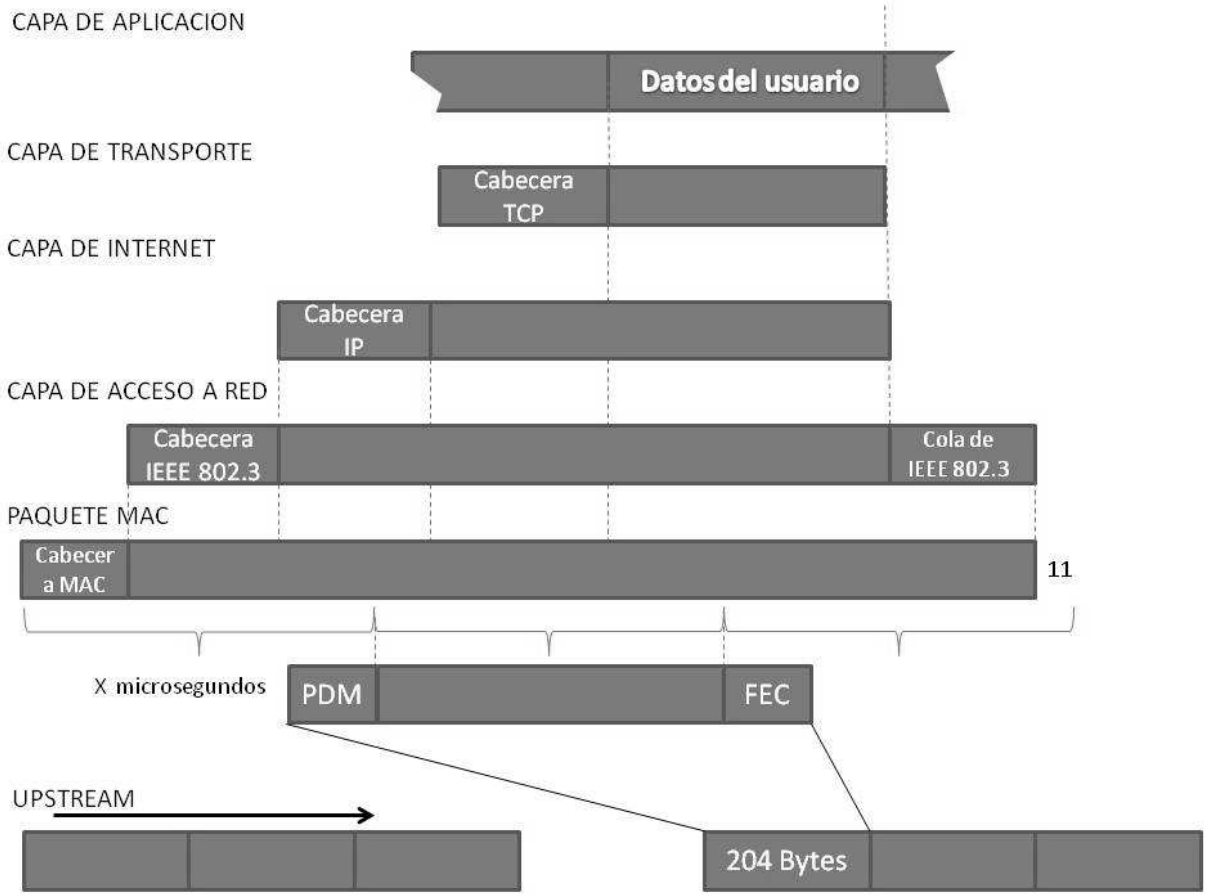


Figura 85

7.7-ATM

El Modo de Transferencia Asíncrona ATM (*Asynchronous Transfer Mode*) es una tecnología de telecomunicación desarrollada para hacer frente a grandes demandas de capacidad de transmisión. Si bien hoy están naciendo nuevas tecnologías, la mayoría de las redes WAN públicas y privadas que demandan gran capacidad de transmisión son ATM. La versatilidad de la conmutación de paquetes de longitud fija que ofrece ATM es su gran fortaleza. ATM se basa en la multiplexación de celdas (así se llama a los paquetes de datos en ATM) de longitud fija (53 bytes) y es una tecnología orientada a la conexión. ATM fue construido para soportar la gran demanda de transmisión de datos de las redes de banda ancha. Fundamentalmente, se pensó para la transmisión de diferentes tipos de información, incluido el video, la voz y los datos. ATM puede intercalar este tipo de información y trasmitirla a velocidades que pueden ir desde 155 Mbps a 50 Gbps. Las redes ATM tienen su propia forma de envío y enrutamiento de paquetes con su propio manejo de

direcciones. ATM surgió en los años 80 alejado de la familia TCP/IP, y hasta cierto punto podían verse como protocolos competidores. ATM fue impulsado fuertemente por las grandes compañías de telefonía. Con el correr del tiempo y la gran difusión de TCP/IP, estos protocolos se han ido acercando para funcionar integradamente.

La red ATM está formada por *switches* también llamados conmutadores (Ver Figura 86). Hay *switches* preparados para realizar interfaces con los usuarios, estas interfaces se denominan UNI (*User-Network Interface*). Son equipos que tienen la capacidad de adaptarse a diversas tecnologías de entradas-salidas (por ejemplo, Ethernet). Luego, están los *switches* para intercambiar celdas ATM exclusivamente. Estas interfaces se denominan NNI (*Network-Network Interface*). Existen UNI y NNI para redes privadas y para redes públicas.

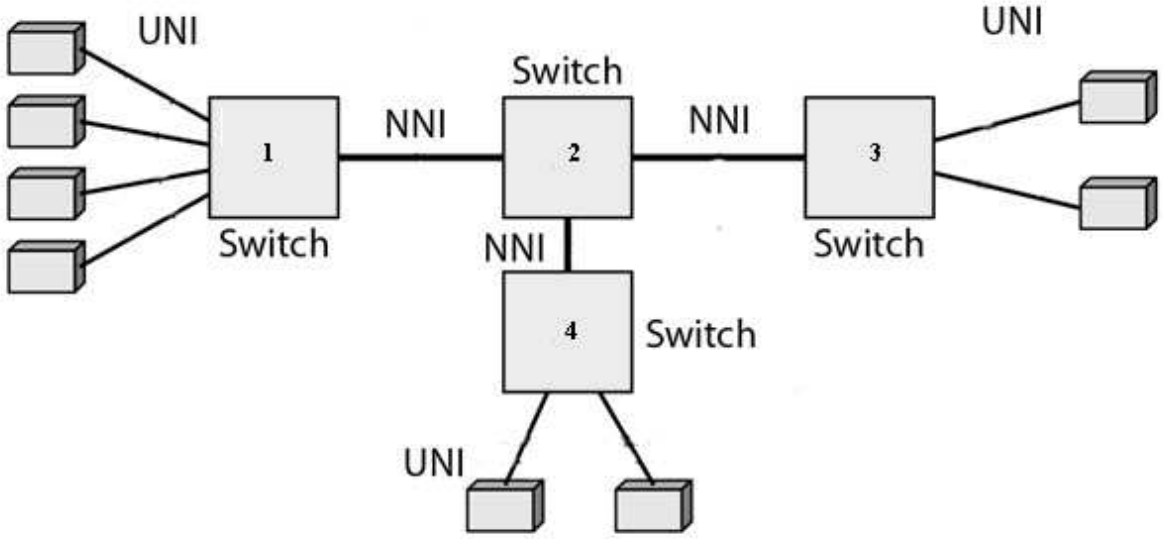


Figura 86

Como ya se comentó, las celdas ATM son de 53 bytes. Toda información que reciban las interfaces UNI será convertida a porciones de 48 bytes a los que se le agregará una cabecera ATM de 5 bytes. En estos encabezados, entre otras cosas, se ingresa la dirección VPI (*Virtual Path Identifier*) y/o la dirección VCI (*Virtual Path Identifier*) que se utilizan para dirigir las celdas a su destino (Ver Figura 87); luego se verán sus funciones específicas. Para permitir la comunicación de datos a alta velocidad, la conexión entre los *switches* se realiza, generalmente, por medio de fibra óptica.

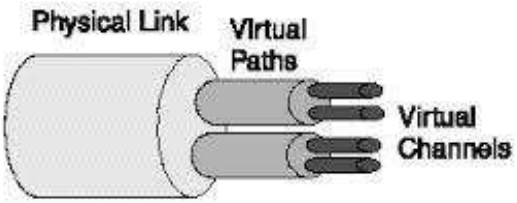


Figura 87

En ATM un enlace entre dos puntos empieza cuando un dispositivo transmite una solicitud a través de la UNI a la red. Mecanismos de señalización pasan la señal a través de la red a su destino. Si el sistema indica que se acepta la conexión, un circuito virtual es establecido a través de la red ATM entre los dos puntos. Cada uno de los *switches* de la red que intervienen en la comunicación determina, en sus tablas internas, cómo tratar cada celda. Los *switches* ATM establecen conexiones punto a punto con el switch inmediato siguiente, de tal forma que el medio no es disputado y, por lo tanto, no se pueden producir colisiones, aunque sí pueden verse saturados por la demanda.

7.7.1-VCI y VPI

ATM maneja, para enrutar las celdas, dos campos (valores): VCI (identificador de circuito virtual, del inglés *Virtual Circuit Identifier*) y VPI (identificador de camino virtual, del inglés *Virtual Path Identifier*), lo que permite dividir el flujo de celdas en dos niveles: el de VC (canal virtual) y el VP (camino virtual). Los paquetes con el mismo valor de VPI, pero diferentes valores de VCI, son enrutados a través de los mismos nodos de la red (un VPI es un conjunto de canales virtuales que atraviesan multiplexadamente un mismo tramo de la red ATM). La existencia de estos dos valores se justifica para ahorrar tiempo en la búsqueda de elementos dentro de las tablas que contienen las relaciones entre los puertos físicos y los VPI/VCI.

Como ya se mencionó anteriormente, ATM funciona como un servicio orientado a la conexión, por lo que con antelación a la transferencia de datos se debe establecer un recorrido (que puede implicar uno o varios *switches* intermedios) y, también, se deben reservar los recursos que la red necesita para transportar la información. Tanto el VPI como VCI tienen existencia localmente, lo que significa que cada *switch* crea sus propios VCI y VPI.

Por lo expuesto, se puede identificar que los *switches* ATM manejan dos niveles de conmutación. El nivel de conmutación de celdas en función del análisis del VPI y el nivel de conmutación de celdas en función del análisis del VCI.

La figura 88 muestra una conmutación a nivel de VPI:

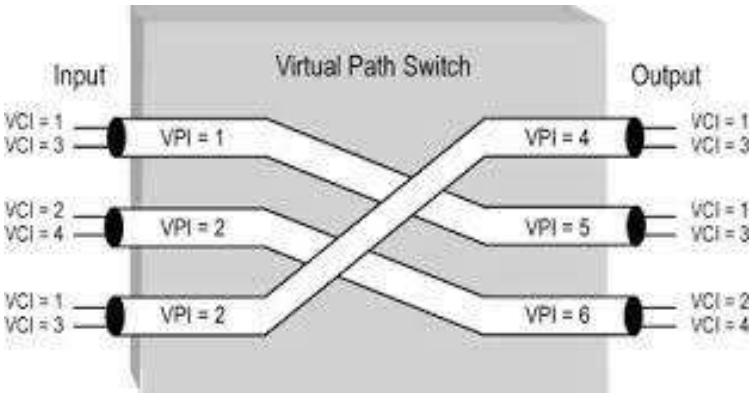


Figura 88

En el siguiente gráfico (Ver Figura 89), se puede observar un ruteo de un modelo sencillo sobre la base de la conmutación a nivel VPI. Se observa la simplificación de los *switches* Z y W que operan únicamente a nivel VPI.

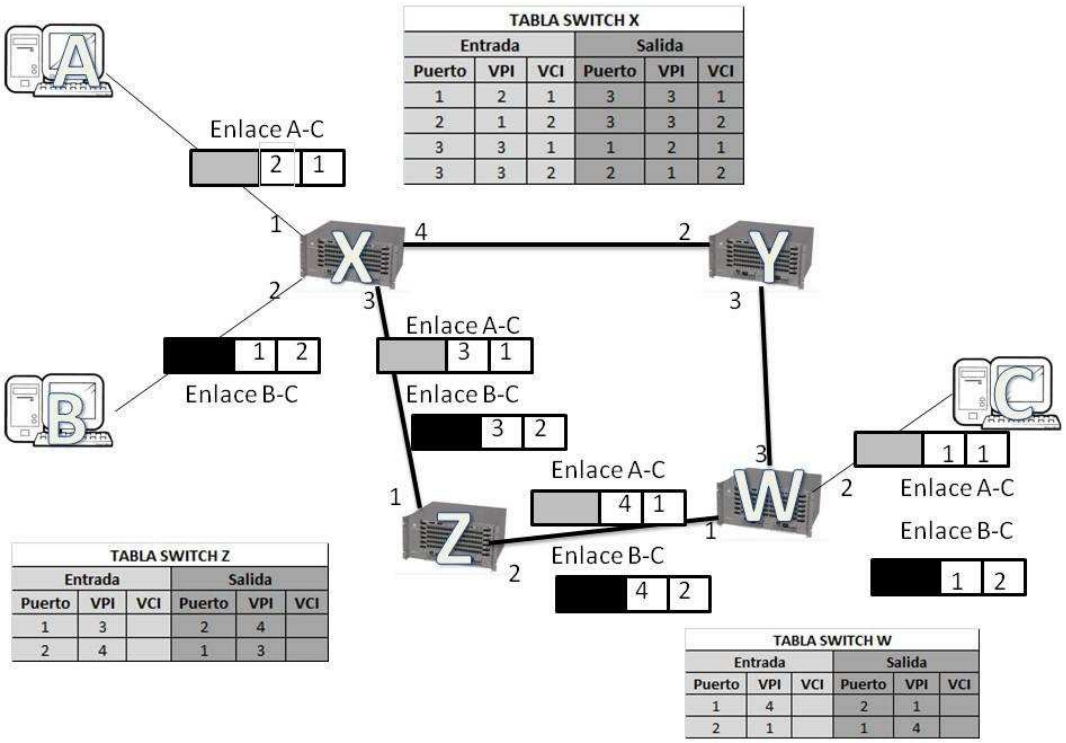


Figura 89

En cada *switch* existen tablas de una base de datos que contiene las relaciones entre los puertos del *swith* y los VCI/VPI (tanto de entrada como de salida).

En la figura 89, se puede ver dos conexiones: una que conecta la computadora A con la C (celdas gris claritas) y otra que conecta la computadora B con la C (celdas negras). Mediante un protocolo denominado PNNI, al establecerse la conexión entre las dos puntas del canal, los *switches* intervinientes crean en sus tablas el enrutamiento de las celdas. En el *switch* **X** se puede ver la tabla de enrutamiento con los caminos bidireccionales creados para las dos conexiones. Por el puerto 1 ingresa la celda con VPI/VCI 21 que sale por el puerto 3 con VPI/VCI 31, es decir que el *switch* cambia el VPI 2 de la celda de entrada por el VPI 3 para la celda en la salida y mantiene sin modificar el VCI. En el puerto 3 de este mismo *switch*, se puede ver como entrada el camino inverso de la conexión anterior. Si el lector sigue con atención la figura, podrá entender el resto de las tablas de los demás *switches*.

A continuación, en la figura 90, se puede ver un *switch* conmutando a nivel de VPI y VCI.

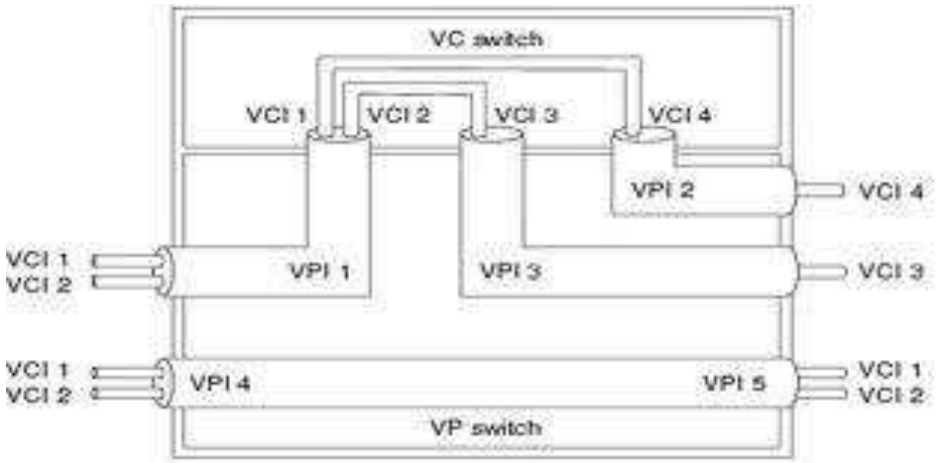


Figura 90

En esta figura, se puede observar un nivel de VP *switch* y otro de VC swith. Este es uno de los aspectos más potentes del *switch* ATM, ya que puede manejarse a cualquier de los dos niveles en función de la complejidad de ruteo que presente la red. Se puede observar que el VPI 1 se conmuta con los VPI 2 y 3. En este caso, no solo se conmuta el VPI, sino que además el VCI 1 del VPI 1 se conmuta por el VCI 4 del VPI 2 y el VCI 2 del VPI 1 conmuta con el VCI 3 del VPI 3. El caso del VCI 1 y 2 del VPI 4 que conmuta a nivel VPI con el VPI 5 es muy similar al caso visto en la figura 89.

En la figura 91, se muestra un ruteo que en todos los casos se conmuta a nivel de VCI. Cuando el VCI de una comunicación cambia entre la entrada a un *switch* y la salida de ese mismo *switch*, las tablas internas deben manejarse al nivel de VPI/VCI.

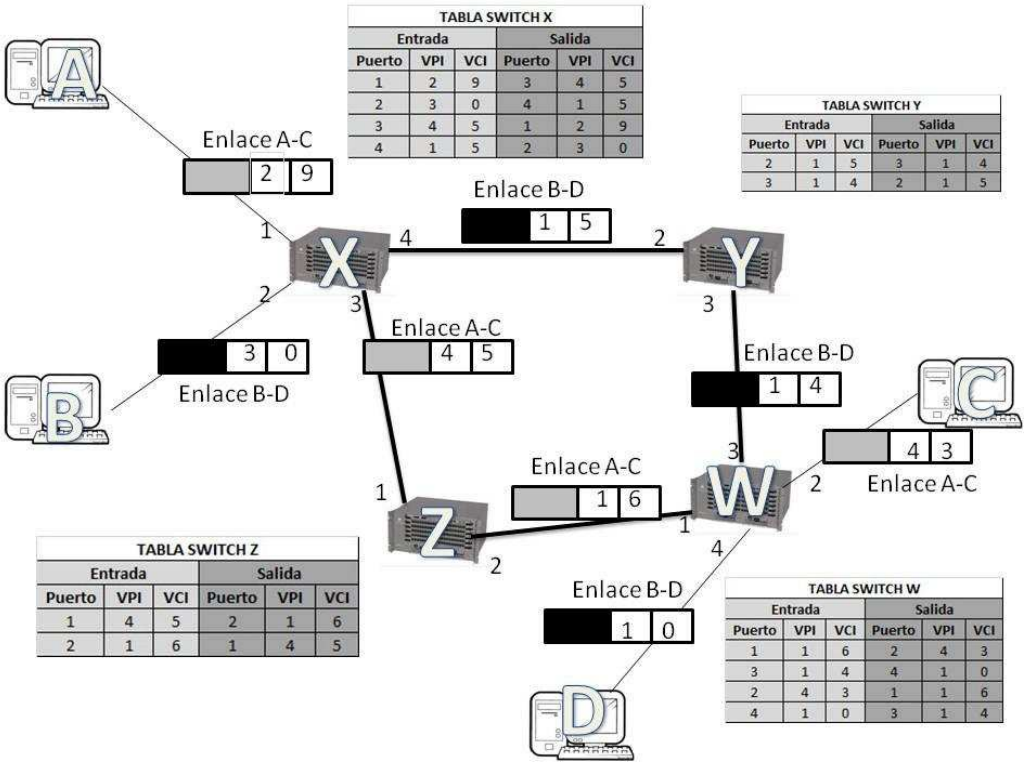


Figura 91

En la red ATM puede haber un gran número (millones) de VCI compartiendo el mismo enlace. De acuerdo con el procedimiento anterior, cada uno tendría VCIs diferentes y, al llegar a un nodo intermedio, habría que consultar tablas de una envergadura considerable. Utilizando las propiedades de agrupar los VCIs en VPIs, estas tablas disminuyen su volumen de información significativamente.

7.7.2-SVC y PVC

Las tablas de los *switches* ATM pueden ser alimentadas por los protocolos de enrutamiento y tienen la característica de ser dadas de altas cuando se inicia la conexión y borradas a su finalización. A esta forma de gestión de las conexiones se la denomina Conexiones Virtuales Conmutadas o SVC (*Switched Virtual Connection*).

Cuando los operadores quieren asegurar rutas fijas para determinados clientes (a los efectos de asegurar velocidades y brindar mayor confiabilidad), lo hacen dando el alta en forma manual los VPI/VCI en las tablas de los *switches* correspondientes. A estas conexiones se las denomina como Conexiones Virtuales Permanentes o PVC (*Permanent Virtual Connection*).

7.7.3-Cabeceras ATM

Las cabeceras de las celdas ATM difieren si se trata de una interfaz UNI o una NNI. En la figura 92, se muestra una celda UNI y otra NNI con sus respectivas cabeceras.

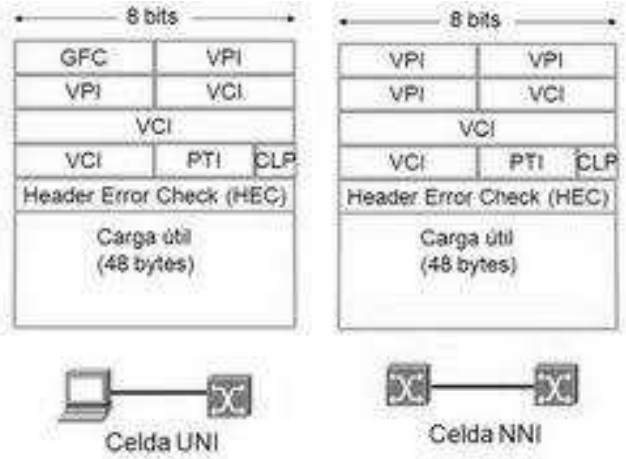


Figura 92

a) GFC: *Generic Flow Control*: 4 bytes en UNI, no está presente en NNI.
b) VPI: *Virtual Path Identifier*: 8 bits en UNI, 12 en NNI.
c) VCI: *Virtual Circuit Identifier*: 16 bits.
d) PT: *Payload Type*: 3 bits.
e) CLP: *Cell Loss Priority*: 1 bit.
f) HEC: *Header Error Control*: 8 bits (CRC sobre los 5 bytes de cabecera).

El aspecto más interesante al ver estas cabeceras es la utilización de los VPI y VCI. Cuando las tablas de los *switches* trabajan con los niveles VPI y VCI, deben manejar direcciones de 8 o 12 bits más 16 bits del VCI, lo que generan tablas de millones de posibilidades. Mucho menos es si el *switch* puede manejarse con los 8 o 12 bits del VPI solamente.

7.7.4-Multiplexación ATM

ATM multiplexa muchas celdas en cada uno de los CV y las coloca en particiones de tiempos (*slots*), esto es similar a una multiplexación por división de tiempo (TDM). Sin embargo, ATM llena cada *slot* con celdas de un CV a la primera oportunidad, de forma similar a la operación de una red conmutada de paquetes (Ver Figura 93). Esto significa que el primer paquete que llega es el que entra en el primer *slot* vacío que se le presenta para ocupar (se suele llamar como Primero Entrado Primero Salido). Los *slots* de celda no usados son llenados con celdas «idle», identificadas por un patrón específico en la cabecera de la celda.

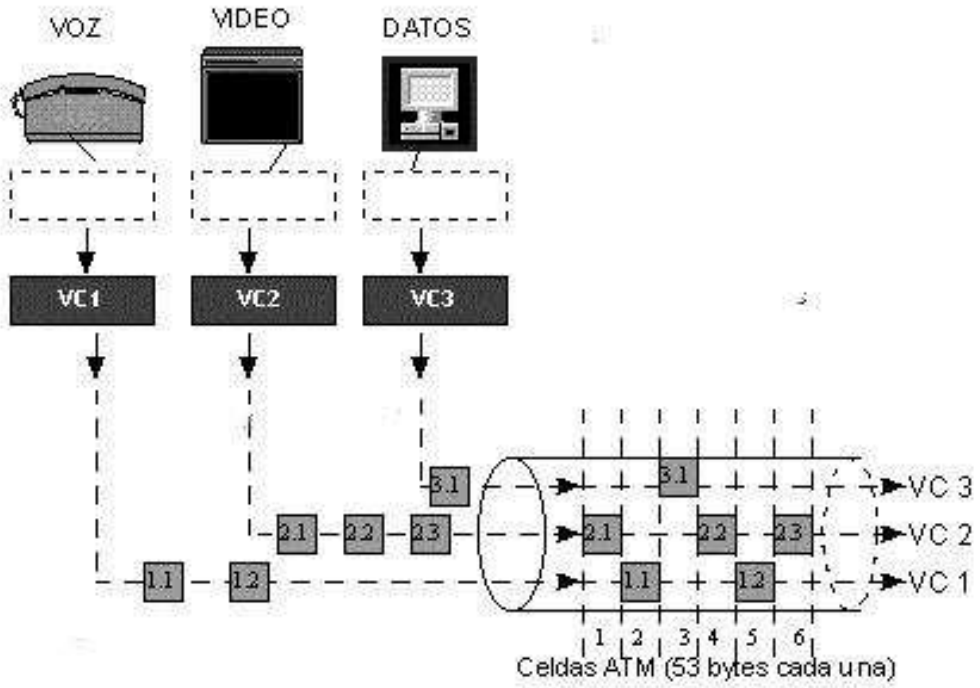


Figura 93

7.7.5-Las tres capas principales de ATM

Lo expuesto anteriormente se puede describir en las tres capas básicas en que se estructura el protocolo ATM:

a) La capa física
b) La capa ATM
c) La capa de adaptación (AAL)

La capa de adaptación

Esta capa se denomina *ATM Adaptation Layer* (AAL). Juega un rol fundamental en la adaptación de ATM a los distintos tipos de servicios. Básicamente, diferencia aquellas aplicaciones que necesitan un flujo constante de bits, como lo son las trasmisiones telefónicas y de video, de aquellas en que pequeños retardos en el flujo de bits no influyen en la calidad del servicio brindado (páginas web).

Su trabajo específico es adaptar los servicios requeridos por las capas más altas, tales como video, audio, textos, etcétera, a las posibilidades de la capa ATM. Esta capa recibe los datos de varias aplicaciones y las convierte en paquetes de 48 bytes que luego serán entregados a la capa ATM. Actualmente están definidos cuatro tipos de servicios AAL.

Esta capa se subdivide en dos capas: Subcapa de Convergencia (CS, *Convergence Sublayer*) y la Subcapa de Segmentación y Reensamblado (SAR, *Segmentation And Reassembly sublayer*) (Ver Figura 94).

La capa CS toma el paquete entregado por la capa de aplicación y crea los paquetes CS-PDU (*Protocol Data Units*) agregando una cabecera y una cola a cada paquete recibido. La información en la cabecera y la cola del paquete depende de la clase de información que va a ser transportada.

La subcapa SAR recibe los paquetes CS-PDU de la capa CS y los subdivide en paquetes de 48 bytes a los que se denominan SAR-PDU (*Segmentation And Reassembler Protocol*). Esta capa agrega su propia cabecera y cola a cada paquete. Por último, estos serán entregados a la capa ATM.

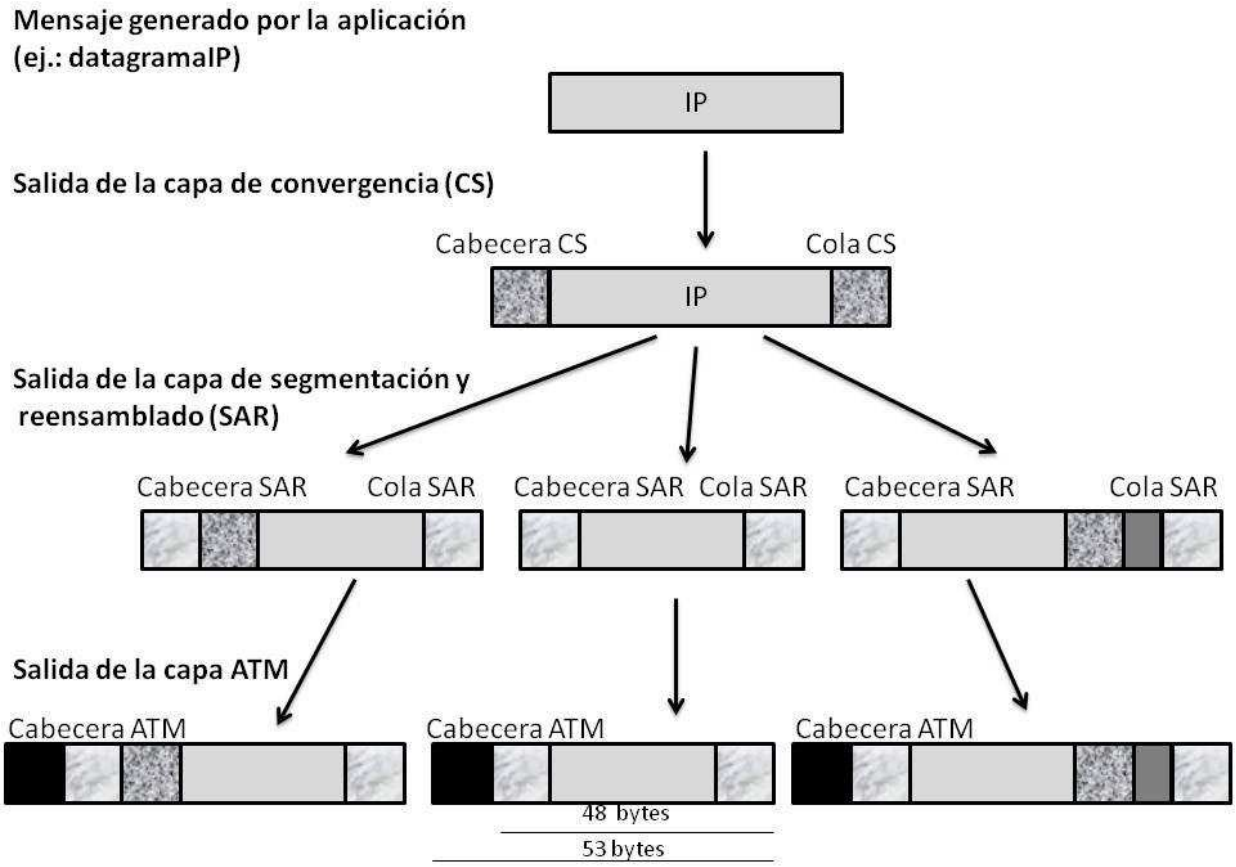


Figura 94

Ambas capas también realizan la función inversa.

La capa ATM

En esta capa, se define la estructura de la celda ATM y cómo las celdas fluyen sobre las conexiones lógicas en la red ATM. Como ya se dijo, el formato de una celda ATM

es muy simple, consiste de 5 bytes de cabecera y 48 bytes para información. Las celdas son transmitidas serialmente y se propagan en estricta secuencia numérica. El tamaño de la celda ha sido escogido como un compromiso entre largas celda que son muy eficiente para transmitir largas tramas de datos y longitudes de celdas cortas que minimizan el retardo de procesamiento de extremo a extremo (que son buenas para voz y video). Los comités de estándares han definido dos tipos de cabeceras ATM: UNI y NNI ya descriptas.

La capa física

La capa física define las interfaces físicas con los medios de transmisión. Es responsable de la correcta transmisión y recepción de los bits. A diferencia de muchas tecnologías que especifica medios de transmisión (por ejemplo, 10 base T, 10 base 5, etc.), ATM es independiente del transporte físico. Las celdas ATM pueden ser transportadas en redes SONET (*Synchronous Optical Network*), SDH (*Synchronous Digital Hierarchy*), T3/E3, TI/EI o aun en módems de 9600 bps. En esta capa se especifican detalles que tiene que ver con velocidades de transmisión, tipos de conectores físicos, incluye la generación y el chequeo del *Header Error Correction* (HEC), reconocimiento del límite de la celda y gestión de reloj.

7.7.6-Clases de servicio AAL

ATM establece cuatro tipos de servicios básicos en función de tres parámetros: la relación de temporización entre origen y destino, el modo de conexión y la velocidad binaria (Ver Figura 95).

Parámetros	**Clase A**	**Clase B**	**Clase C**	**Clase D**
Temporización	Requerida		No requerida	
Velocidad binaria	Constante	Variable		
Modo de conexión	Orientada a la conexión			Sin establecer conexión
Ejemplo	Video/Audio	Video	Datos	Datos

Figura 95

En función de las variantes que permiten los tres parámetros antedichos, se crearon 4 tipos de protocolos AAL: AAL1, AAL2, AAL3/4 y AAL5.

AAL1

La AAL1 (la clase A) se usa para transferir tasas de bits constantes que dependen del tiempo. Provee la recuperación de errores e indica la información con errores que no podrá ser recuperada.

La cabecera SAR de este servicio es de 1 byte que contiene:

- Número de secuencia de 4 bits
- Código de protección de errores de 4 bits

La capa CS maneja, entre otras cosas, la tolerancia al retardo de celdas, efectúa una sincronización entre el reloj de salida y el reloj de recepción, monitorea células perdidas y realiza corrección de errores. Puede agregar o no datos de cabecera y cola.

AAL2

Esta capa está definida para brindar servicios de clase B. Como se puede observar, la diferencia con AAL1 es que debe adaptarse a una velocidad de transferencia variable, fundamentalmente, para adaptarse a la compresión de video MPEG. Los formatos de los paquetes CS y SAR son similares a los de la capa AAL1.

AAL3/4

Estas variantes dan respuestas a las clases de servicios C y D. Actualmente prácticamente en desuso.

AAL5

Estas variantes dan respuestas a las clases de servicios C y D. Por sus características y beneficios, se ha impuesto su uso antes que el protocolo AAL3/4. Es un servicio pensado para la transmisión de datos que no requieren una tasa de transferencia constante ni temporización.

La capa CS agrega una cola con la siguiente información (Ver Figura 96):

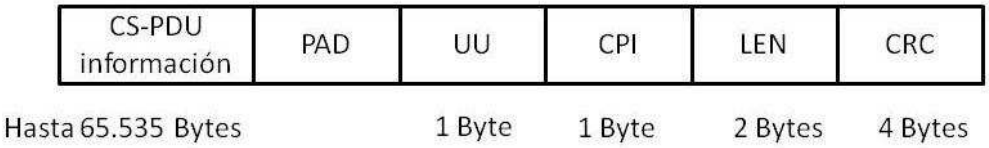


Figura 96

- PAD: Es un espacio de relleno para llevar el total del paquete a un múltiplo de 48 bytes.
- UU: Actualmente no se usa.
- CPI: Es el indicador de parte común que posibilita la interpretación del resto de los campos y el alineamiento de la cola a 8 bytes
- LEN: Es la longitud total del *payload*, para saber la carga útil.
- CRC: Es el código de redundancia cíclica para detección de errores. Se calcula sobre el total del paquete CS.

El paquete SAR no agrega información.

7.7.7-Direcciones

Hasta aquí se ha visto, básicamente, cómo trabaja ATM sobre la base de la conmutación y el enrutamiento vía la utilización de VPI y VCI. Previo a que ATM pueda establecer el esquema de conmutación y ruteo entre una máquina emisora y otra receptora, la emisora debe poder localizar el dispositivo receptor. Para este fin ATM utiliza direcciones propias que son distintas de las direcciones IP. Tal cual el caso de las direcciones MAC en las redes Ethernet.

ATM utiliza el estándar de direcciones E.164 (es el conocido formato telefónico de 15 dígitos) para redes públicas, que es un estándar de la Unión Internacional de Comunicaciones (UIT) y NSAP para redes privadas administradas por el ATM Forum.

Las direcciones NSAP están definidas en la norma ISO 8348 y la recomendación ITU-T X.213. Existen tres formatos diferentes para direcciones NSAP: DCC, ICD, E.164 (todas ellas de 20 bytes). Los tres formatos se pueden dividir en dos partes principales que son: 1) el identificador de *switch* y 2) la dirección MAC.

En el identificador de *switch*, los primeros 13 bytes identifican un *switch* particular en la red ATM. El uso de esta porción de la dirección puede variar considerablemente dependiendo de cuál de los tres formatos de dirección está en

uso. La dirección MAC son 6 bytes que identifican un punto final físico, tal como un adaptador ATM específico. El uso y la asignación de las direcciones MAC de hardware ATM son idénticos a la dirección MAC que se utiliza en Ethernet.

Los tres formatos NSAP se puede observar en la figura 97.

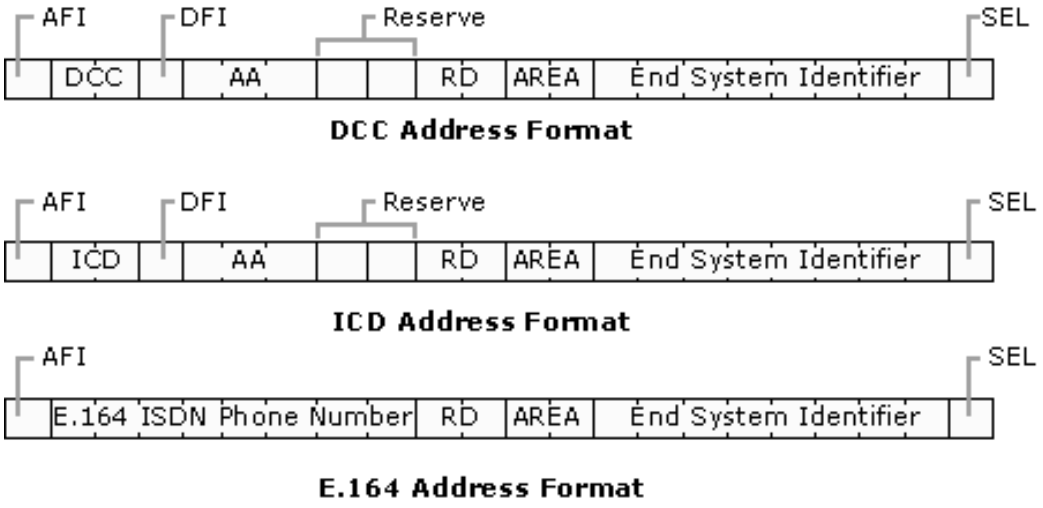


Figura 97

La siguiente tabla especifica el significado de cada campo.

Campo	Función
AFI	Su valor es 45 para direcciones E.164, 47 para ICD y 39 para DCC.
DCC	Identifica un determinado país de acuerdo a la norma ISO 3166.
DFI	Especifica la administración, semántica y estructura del resto de la dirección.
AA	Identifica organizaciones como proveedores de servicios ATM o administradores de redes ATM privadas.
Reserve	Reservado para futuros usos.
RD	Especifica un dominio dentro de un único segmento con idéntico E.164, DCC/DFI/AA o ICD/DFI/AA.
Area	2 bytes de área de identificación.

ESI	*End System Identifier*. Especifica una dirección MAC, identifica una terminal de ATM dentro de un área específica.
SEL	No se utiliza para el enrutamiento y no tiene significado dentro de la red ATM, pero puede ser utilizado para multiplexar información en la terminal de ATM cuando se comparte una interfaz de ATM.
ICD	Identifica una determinada organización internacional. El encargado del registro de este código es el *British Standard Institute*.
E.164	8 bytes (15 caracteres) para el número telefónico ISDN.

7.7.8-Ruteo en ATM

En el mercado, existen varios protocolos que permiten que cada *switch* arme su tabla de ruteo (es decir, su encaminamiento hacia los distintos destinos, construyendo para cada sección su correspondiente VPI/VCI). Uno de los más utilizados en redes privadas es el denominado *Private Network-Network Interfaz* (PNNI). En este protocolo, cada *switch* envía información relativa a sus vecinos en forma de *broadcast* a todos los *switches* que componen la red. La diferencia fundamental con RIP es que este último envía información a sus vecinos y estos a sus vecinos. Además, RIP utiliza un algoritmo denominado de vector-distancia (*distance-vector*), donde el costo de cada enlace siempre es uno (cada salto de un *switch* a otro vale 1). En cambio, para determinar el costo de cada ruta, PNNI utiliza el tipo de algoritmo denominado estado del enlace (*link-state*), donde el costo de cada enlace no tiene que ser siempre 1, sino que además se consideran otros elementos, como el ancho de banda.

7.7.9-IP sobre ATM

Existen varios esquemas de trabajo de IP sobre ATM para redes WAN. A continuación se verán los más importantes:

CLIP (*Classical IP over ATM*)

En su forma más sencilla, IP viaja sobre redes ATM bajo la norma RF 1577 (comúnmente llamada CLIP), donde ATM se utiliza tan solo como un medio de transporte punto a punto y encapsulado en un PDU del AAL5. No se aprovecha ninguna de las posibilidades de ATM, ni siquiera la posibilidad de brindar calidad de

servicio. Bajo este esquema, la red ATM se subdivide en subredes lógicas IP (*Logical IP Subnet*, LIS) (Ver Figura 98).

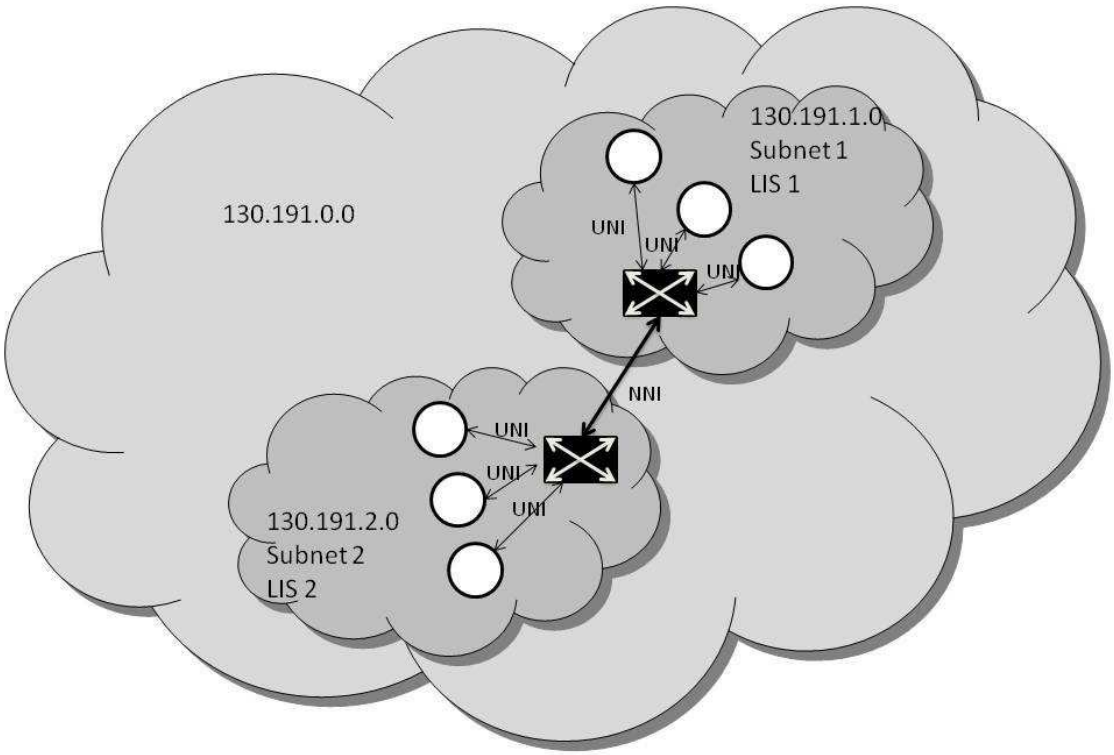


Figura 98

Dentro de cada LIS, el protocolo ATM ARP se encarga de convertir las direcciones IP en direcciones ATM MAC (esto se realiza utilizando un servidor que proporciona el servicio de ATM ARP) (Ver Figura 99).

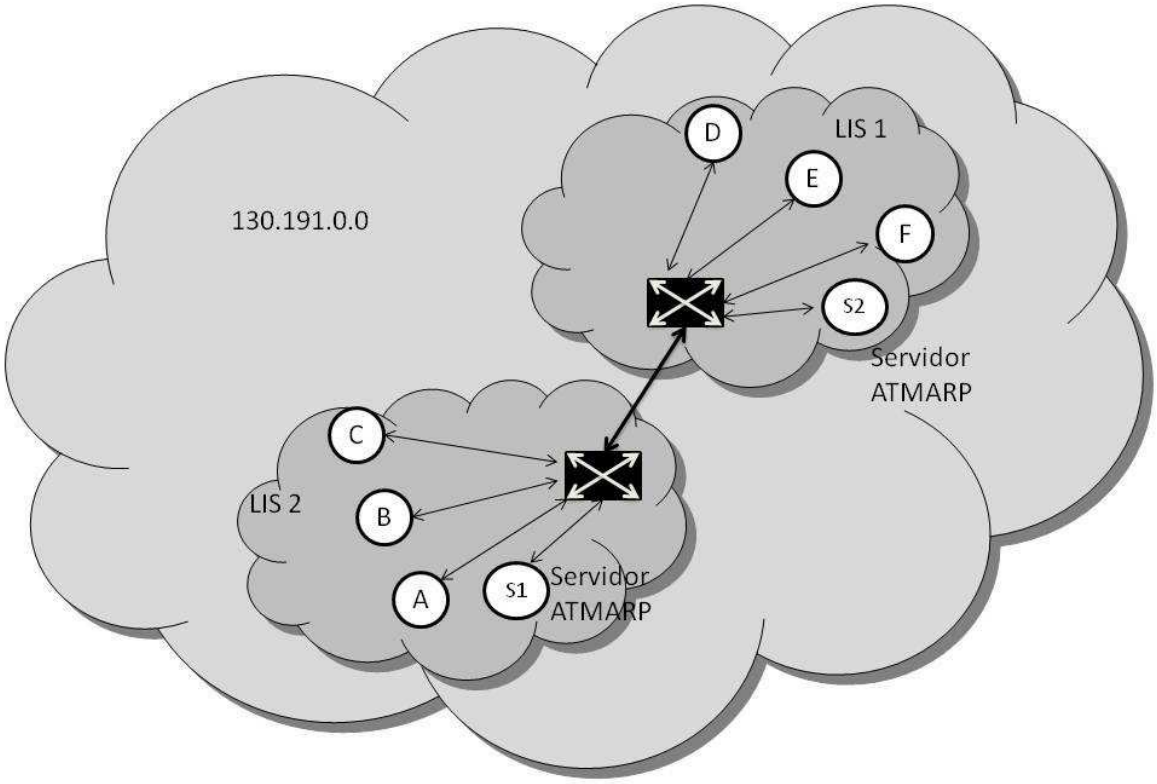


Figura 99

Cuando un *host* desea comunicarse con otra dentro de la misma LIS, primero debe convertir la dirección IP en la dirección ATM MAC, que es el ESI de cualquiera de los formatos de direcciones NSAP, para luego invocar la conexión a esa ESI de la red ATM, a partir de la cual opera el sistema de ruteo de la propia red ATM creando las correspondientes VCCs (*Virtual Channel Conections*).

Cuando un paquete IP debe pasar información a un *host* que está en otra LIS, debe realizarse una compleja operación, que es la mayor desventaja de este modelo. Una LIS está unida a otra LIS por *switches* ATM, entre los cuales se establece un VCC en forma manual. Cuando un *host*, por ejemplo el A de la LIS 2 (Ver Figura 100), desea pasar un paquete IP al *host* D de la LIS 1 primero debe buscar en el servidor S1 para encontrar su dirección ATM MAC. Como S1 no tiene registrada la IP del *host* D de la LIS 1, debe pasar el paquete a la LIS 2 por intermedio de la interfaz NNI. Cuando el *switch* de la LIS 1 recibe las celdas ATM que contiene el paquete IP, primero debe reconstruir el paquete IP, luego obtener la dirección ATM MAC en el servidor S2 para luego volver a encapsularlo en un PDU del AAL5 y establecer el ruteo de las nuevas celdas hacia el *host* D. Si la estación de destino no estuviera en la LIS 2, deberá repetirse el procedimiento en cada LIS de la red hasta encontrar la estación buscada.

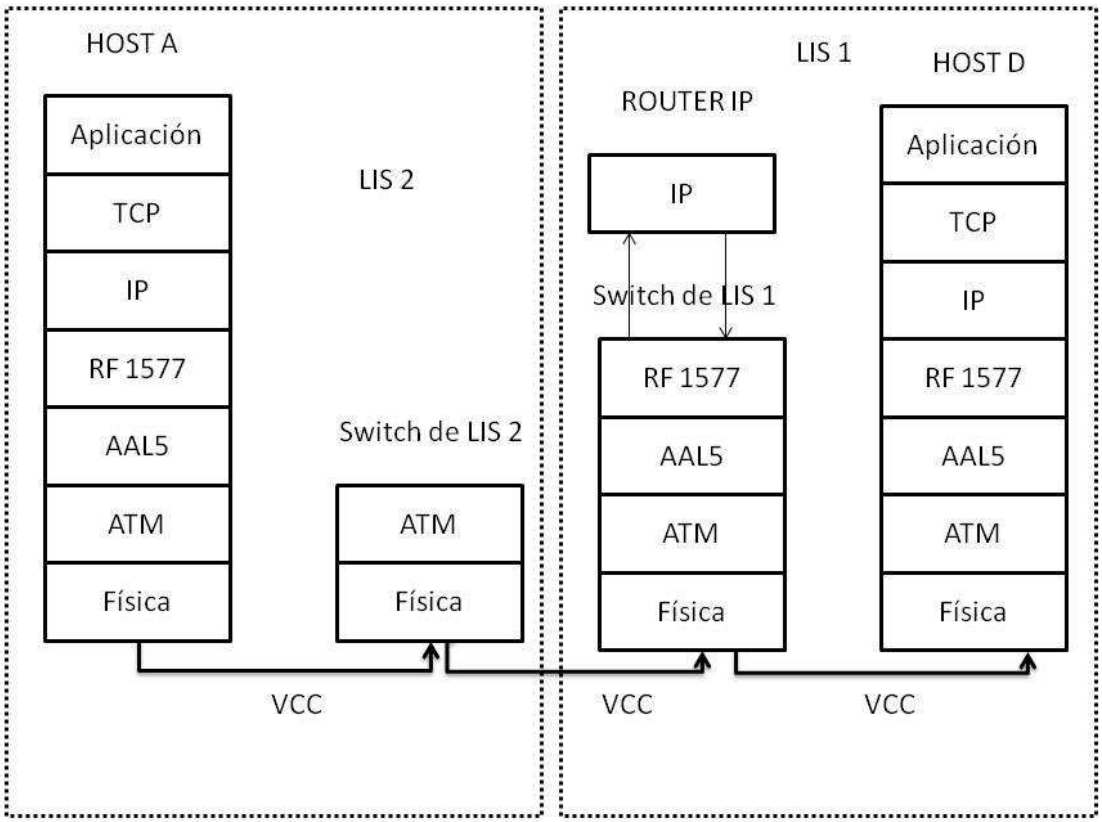


Figura 100

NHRP (*Next Hop Resolution Protocol*)

Para que dos estaciones ATM que pertenecen a LIS distintas se puedan conectar, el IETF incorporó el mecanismo denominado *Next Hop Resolution Protocol* (NHRP). Este protocolo es más avanzado y permite interconectar estaciones de diferentes LIS sin necesidad de pasar por el tedioso proceso de reconstruir el paquete IP y volver a generar las celdas ATM en cada paso de una LIS a otra.

En este modelo, el Servidor de direcciones ATM MAC tiene la capacidad de comunicarse con los servidores de de las otras LIS (utilizando circuitos virtuales permanentes) para encontrar el *host* de destino, de tal manera que desde el *host* origen el paquete IP es encapsulado en un PDU de AAL5 y enviado en celdas ATM hasta el *host* de destino sin necesidad de efectuar el proceso de volver al paquete IP cada vez que se pasa de una LIS a otra.

7.7.10-Mejoras de hardware para soluciones IP sobre ATM

Se han desarrollado varias tecnologías de hardware para desarrollar altas velocidades de transmisión en redes IP sobre ATM. La más destacada es la de matriz de conmutación del tipo barras cruzadas (*crossbar*). En estas tecnologías, se reemplaza el bus de interconexión de las interfaces de red por matrices de conmutación del tipo barras cruzadas. Esto permite conmutar paquetes en paralelo, mientras que el bus solo permite conmutar un paquete a la vez. La utilización de hardware específico para determinar la dirección IP de cada paquete y buscarla en las tablas de encaminamiento permite acelerar notablemente el procesamiento de los paquetes IP. Esta tecnología se fundamenta en lograr el encaminamiento de los paquetes de información fundamentalmente con hardware, en lugar de realizarlo con software.