



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE ESTUDIOS SUPERIORES  
ARAGÓN

**RESUMEN DEL DIPLOMADO DE TECNOLOGIAS  
DE INFORMACIÓN TERCERA EDICIÓN**

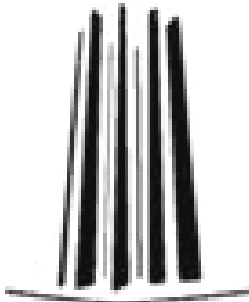
SEMINARIOS Y CURSOS DE ACTUALIZACIÓN  
Y CAPACITACIÓN PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN  
P R E S E N T A:

O S C A R S E V I L L A J U Á R E Z

A S E S O R D E T E S I S :  
M. EN C. LEOBARDO HERNÁNDEZ AUDELO

SAN JUAN DE ARAGÓN, ESTADO DE MEXICO 2007





Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## **Agradecimientos**

---

Quiero agradecer primeramente a Dios que me permitiera terminar una carrera profesional.

Enseguida agradezco a mi Hija Tania Paola quien fue mi motor, fuerza e inspiración para que a pesar de diversas adversidades pudiera al fin terminar este trabajo.

A mi Esposa Elsi que a su modo me dio el coraje.

Sin duda a mis Padres Yolanda Juárez y José Félix Sevilla quienes sin ellos no sería nada y no estaría presente para poder escribir estos párrafos.

A mis Hermanos Cesar y Edgar por su paciencia y apoyo.

---

---

## Convenciones Tipográficas

Las convenciones en el presente documento se encuentran descritas en la siguiente tabla mostrando del lado izquierdo una breve descripción de la convención y del lado derecho mostrando como ejemplo el formato utilizado.

Descripción de la convención	Ejemplo del Formato
Los capítulos principales se mostrarán en negrita, con un tipo de letra Arial de tamaño 14 y alineación centrada.	<b>Capítulos Principales</b>
Los subtítulos de cada capítulo se mostrarán en negrita, con un tipo de letra Arial de tamaño 13 y alineación izquierda.	<b>Subtítulos</b>
Los títulos importantes que se encuentran inmersos en los capítulos y subtítulos se mostrarán en negrita, con un tipo de letra Arial de tamaño 12 y alineación a la izquierda.	<b>Títulos importantes</b>
Para poder resaltar algunas definiciones en los listados o al inicio de párrafos se utilizará un tipo de letra Arial en negrita con tamaño 11 y alineación justificada.	<b>Definiciones en listados</b>
Para resaltar palabras o frases dentro del texto que represente algo importante, se utilizará como tipo de letra Arial en cursiva y subrayada.	Palabra a <u><i>resaltar</i></u> dentro del texto
Los nombres de las figuras se colocarán con Arial de tamaño 9, centrada, con borde superior y se tipificará de la siguiente manera: la palabra "Figura X." será en negrita, y la descripción o nombre la figura será sin resaltar.	<hr/> <b>Figura X.</b> - Nombre de la figura

---

## Lista de Figuras

	<b>Página</b>
Figura.- 1 Modelo de Von Neumann.....	4
Figura.- 2 Bloques Funcionales Von Neumann.....	4
Figura.- 3 Estructura Básica de un Procesador.....	9
Figura.- 4 Sistema Uniprogramado.....	15
Figura.- 5 Multiprogramación.....	15
Figura.- 6 Sistemas Operativos por Servicios.....	16
Figura.- 7 Sistemas Operativos (1950 – 2006).....	16
Figura.- 8 Trama Ethernet.....	29
Figura.- 9 Trama IEEE802.3.....	30
Figura.- 10 Nacimiento de TCP/IP.....	38
Figura.- 11 Correspondencia capas modelos TCP/IP y OSI.....	39
Figura.- 12 Algunas características de las capas de TCP/IP.....	39
Figura.- 13 Net ID y Host ID en una dirección IP.....	40
Figura.- 14 Características de la Onda Senoidal.....	43
Figura.- 15 Amplitud Modulada.....	44
Figura.- 16 Frecuencia Modulada.....	44
Figura.- 17 On/Off Keying (OOK).....	45
Figura.- 18 Frequency Shift Keyeng (FSK).....	45
Figura.- 19 Pulse Amplitude Modulation (PAM).....	45
Figura.- 20 Phase Shift Keying (PSK).....	46
Figura.- 21 Clasificación por Extensión de Redes Inalámbricas.....	48
Figura.- 22 Modelo conceptual de un canal de comunicación.....	54
Figura.- 23 Tipos de Amenazas.....	55
Figura.- 24 Asimétrica o de Llave Pública.....	60
Figura.- 25 Ejemplo de Función Hash (1).....	60
Figura.- 26 Ejemplo de Función Hash (2).....	60
Figura.- 27 Ejemplo de Función Hash (3).....	61
Figura.- 28 Ejemplo de Función Hash (4).....	61
Figura.- 29 Proceso de Firma Digital.....	63
Figura.- 30 Firma Digital con resumen Hash.....	63
Figura.- 31 Ejemplo del Modelo Entidad – Relación.....	70
Figura.- 32 Modelo Lógico.....	72
Figura.- 33 Modelo Físico en Sybase.....	73
Figura.- 34 Arquitectura de Inteligencia de Negocios.....	77

---

	<b>Página</b>
Figura.- 35 Organización Conceptual SOA.....	78
Figura.- 36 Cubo de Información.....	79
Figura.- 37 Relación de un Datamining con otras disciplinas.....	81
Figura.- 38 Fases de Extracción de Conocimiento.....	82
Figura.- 39 Tabla de hechos y dimensiones.....	85
Figura.- 40 Relación entre las tecnologías y el tipo de usuario que las utiliza.	86
Figura.- 41 Modelo Lógico.....	88
Figura.- 42 Modelo Físico.....	89
Figura.- 43 Modelo de Cascada Tradicional.....	92
Figura.- 44 Modelo Espiral.....	92
Figura.- 45 Las cuatro fases del marco de trabajo iterativo incremental.....	93
Figura.- 46 La fase de construcción está formada por una serie de mini cascadas.....	93
Figura.- 47 Vista General de RUP.....	98
Figura.- 48 Asociaciones.....	102
Figura.- 49 Tipos de Asociaciones.....	102
Figura.- 50 Iteraciones en Fase de Construcción.....	103
Figura.- 51 Solicitud de página Web tradicional.....	123
Figura.- 52 El servidor llama a un interprete (PERL).....	123
Figura.- 53 Uso de Contenedores o Servlets.....	124
Figura.- 54 Uso de Application Server.....	124
Figura.- 55 Arquitectura Mono-capa.....	126
Figura.- 56 Arquitectura en dos capas (Two-Tier) .....	126
Figura.- 57 Arquitectura en tres capas (Three-Tier) .....	126
Figura.- 58 Esquema de Negocios Electrónicos.....	136
Figura.- 59 Criterios de Alineación de las TI al Negocio.....	138
Figura.- 60 Ciclo de ventas y mercadotecnia.....	139
Figura.- 61 Fases del ciclo de vida de un proyecto.....	140
Figura.- 62 Ciclo de vida de los proyectos de TI.....	141
Figura.- 63 Correspondencia de ciclos de vida de proyectos RUP-PM.....	141
Figura.- 64 Proceso de Iniciación.....	143
Figura.- 65 Matriz de análisis de requerimientos FURPS+.....	148
Figura.- 66 Carta Constitutiva del Proyecto.....	149
Figura.- 67 Procesos de la Planeación de Proyectos.....	150
Figura.- 68 Descomposición de tareas.....	152
Figura.- 69 Diagrama de Red.....	153

---

	<b>Página</b>
Figura.- 70 Matriz de Asignación de Responsabilidades.....	154
Figura.- 71 Procesos de Ejecución del Proyecto.....	156
Figura.- 72 Modelo Lógico.....	157
Figura.- 73 Modelo Físico en Oracle.....	158
Figura.- 74 Curva Típica de Aprendizaje.....	159
Figura.- 75 Curva de Aprendizaje con Soporte Técnico.....	159
Figura.- 76 Procesos Facilitadores.....	159
Figura.- 77 Procesos de Control.....	161
Figura.- 78 Procesos de Cierre.....	162

---

## Lista de Tablas

		<b>Página</b>	
Tabla.-	1	Unidades de Medida.....	5
Tabla.-	2	Código ASCII (American Standard Code for Information Interchange) (0-127).....	6
Tabla.-	3	Código Extendido ASCII.....	6
Tabla.-	4	Código EBCDIC (Extended Binary-Coded Interchange Code).....	7
Tabla.-	5	Modelo de Referencia OSI.....	23
Tabla.-	6	Comparativa de Modelo OSI y DARPA.....	24
Tabla.-	7	Categorías de 10BaseT.....	26
Tabla.-	8	Características medios de transmisión.....	26
Tabla.-	9	Clases IP.....	41
Tabla.-	10	Estándares Inalámbricos WLAN.....	49
Tabla.-	11	Características de la Información.....	51
Tabla.-	12	Servicios de Seguridad y su correspondiente Mecanismo de Seguridad.....	55
Tabla.-	13	Equivalencia en el Paradigma Relacional.....	66
Tabla.-	14	De esta manera se representa una entidad.....	72
Tabla.-	15	Operaciones de una Empresa.....	140
Tabla.-	16	Mecanismo Arquitectónico de tres niveles.....	144

## Lista de Cuadros

		<b>Página</b>	
Cuadro.-	1	Arquitectura de Computadoras.....	8
Cuadro.-	2	Especificaciones IEEE802.3.....	31
Cuadro.-	3	Características de Fast Ethernet ó 100 BaseT.....	32
Cuadro.-	4	Características de Gigabit Ethernet ó 1000 BaseT.....	32
Cuadro.-	5	Clasificación de la Autenticación.....	56
Cuadro.-	6	Clasificación de la criptografía.....	59
Cuadro.-	7	Multimedia por su Función.....	128
Cuadro.-	8	Herramientas de desarrollo Multimedia.....	129
Cuadro.-	9	Tipos de comercio electrónico.....	132



---

# ÍNDICE

	<b>Página</b>
Prólogo.....	I
Convenciones Tipográficas.....	III
Lista de Figuras.....	IV
Lista de Tablas y Cuadros.....	VII
<b>Capítulo 1 Conceptos Básicos y Sistemas Operativos</b>	
1.1 Concepto de Información.....	1
1.2 Origen y Evolución de las Computadoras.....	2
1.3 Unidades de Información y Códigos.....	6
1.4 Organización, Estructura y Arquitectura de Computadoras.....	7
1.5 Origen de los Sistemas Operativos.....	12
1.6 Sistemas Operativos Modernos.....	17
<b>Capítulo 2 Redes, Telecomunicaciones y Cómputo Móvil</b>	
2.1 Conceptos, Comunicaciones y Estándares.....	22
2.2 Redes de Datos.....	24
2.3 Telecomunicaciones.....	36
2.4 TCP/IP.....	38
2.5 Redes Inalámbricas.....	41
2.6 Fundamentos de Radio Frecuencia.....	42
2.7 Tecnologías de intercomunicación.....	46
2.8 Dispositivos de la infraestructura inalámbrica.....	47
2.9 Tecnologías Inalámbricas, Protocolos y Estándares.....	48
<b>Capítulo 3 Seguridad Informática</b>	
3.1 Seguridad de la Información.....	50
3.2 Seguridad Informática.....	52
3.3 Criptología y Criptografía.....	57
3.4 Firma Digital.....	62
3.5 Certificados Digitales.....	64

---

	<b>Página</b>
<b>Capítulo 4 Bases de Datos y Datawarehouse</b>	
4.1 Conceptos y Definición de sistema manejador de bases de datos (SMBD).....	66
4.2 El enfoque relacional.....	69
4.3 Bases de datos Orientadas a Objetos (BDOO).....	73
4.4 Seguridad, Integridad y Confidencialidad en las Bases de Datos.....	74
4.5 Introducción al Datawarehouse.....	75
4.6 Almacenes de Datos.....	79
4.7 Minería de Datos (Data Mining).....	80
4.8 Datawarehouse.....	83
4.9 Aplicación del Método.....	87
 <b>Capítulo 5 Diseño de Sistemas y Tecnologías de Programación</b>	
5.1 Introducción al Análisis de Sistemas.....	91
5.2 UML.....	94
5.3 Fases de RUP.....	97
5.4 Lenguajes de Programación.....	104
5.5 Ingeniería de software y herramientas.....	106
5.6 Modelo de Desarrollo Rápido de Aplicaciones (RAD).....	107
5.7 Paradigma Orientado a Objetos.....	109
 <b>Capítulo 6 Desarrollo de Aplicaciones en Internet Multimedia y Manejo Digital de Imágenes</b>	
6.1 Conceptos Básicos.....	112
6.2 Tecnologías de desarrollo para Internet.....	116
6.3 Análisis de Integración entre servidores Web manejadores de base de datos.....	123
6.4 Arquitecturas.....	125
6.5 Conceptos Básicos de Multimedia.....	127
6.6 Tipos de Sistemas .....	127
6.7 Multimedia por su Función.....	127
6.8 Elementos Multimedia.....	128
6.9 Herramientas de Desarrollo.....	129

---

---

	<b>Página</b>
<b>Capítulo 7 Comercio Electrónico y Administración de Proyectos de TI</b>	
7.1 Introducción al Comercio Electrónico.....	131
7.2 EDI.....	133
7.3 Servicios Web.....	135
7.4 Tecnologías comunes en comercio electrónico.....	136
7.5 Plan de Negocio.....	137
7.6 Contexto de la administración de proyectos.....	140
7.7 Fases de la administración de proyectos de TI.....	142
7.8 Planeación.....	150
7.9 Ejecución.....	155
7.10 Control.....	160
7.11 Cierre.....	162
<b>Capítulo 8 Resultados y Conclusiones</b>	<b>164</b>
<b>Referencias</b>	<b>166</b>

---

## PRÓLOGO

El uso de las tecnologías de información en las organizaciones es tema fundamental para su desarrollo y mejor aprovechamiento de sus recursos materiales y humanos, por otro lado la incursión de ellas ha generado enormes avances en diferentes áreas, como la medicina, administración, cultura, economía, cómputo, etc. Y en general cualquier área que persiga crear, almacenar, procesar, transformar y difundir la información.

Las organizaciones deben percibir que para poder competir en el mercado, es necesario estar a la vanguardia en el uso y aplicaciones de tecnologías de información que hagan que sus procesos sean más eficientes y logren mejorar sus resultados a corto, mediano y largo plazo.

Es por esa razón que cada vez es más frecuente que se hagan importantes inversiones económicas en recursos materiales, pero sobre todo en capacitación y actualización del personal que hace uso de dichas tecnologías. Por supuesto, esto provoca que las organizaciones demanden personal más especializado en todas sus áreas.

En virtud de este escenario los nuevos profesionales se enfrentan a retos tecnológicos cada vez más complejos que requieren el manejo de conceptos y habilidades diversas. Y desde luego, las tecnologías de información demandan personal especializado y con diversidad de conocimientos apoyándose en diferentes herramientas.

El presente trabajo tiene el objetivo de mostrar un panorama de diferentes tecnologías de información, enfocadas a la informática como apoyo a los profesionales que requieren cada vez más conocimientos y especialización en ellas, todo ello con el fin de poder tomar las mejores decisiones que repercutirán en avances, o retrasos para las organizaciones.

Este documento se encuentra organizado en 8 capítulos, en los cuales se abordan las bases fundamentales del cómputo, el uso de las tecnologías para el mejor aprovechamiento de la información y al final tener diferentes opciones de solución para las organizaciones.

En el capítulo 1 se abordarán los conceptos básicos de información, el origen y evolución de las computadoras, mostrando por generaciones los acontecimientos más importantes que se suscitaron para que se dieran importantes logros, por ejemplo, la manera en que se procesa la información en la computadora, así como también las bases de su diseño físico y lógico. También se muestra el software necesario para la administración de todos los recursos de la computadora, es decir, los sistemas operativos, así como su evolución en el tiempo.

---

En el capítulo 2 se describe el tema de las telecomunicaciones, las redes y el cómputo móvil, iniciando con conceptos fundamentales como el modelo de referencia OSI, en seguida se revisan las redes de datos describiendo entre otras cosas: medios de transmisión, topologías, tipos de redes, TCP/IP y redes inalámbricas. También se describen conceptos esenciales de radiofrecuencia y características de las ondas, puntualizando la manera en que viaja la información.

En el capítulo 3 se muestran características sumamente importantes en el tema de seguridad del manejo de información, iniciando con un paseo por el tiempo, explicando los problemas y necesidades que se dieron para que se crearán algoritmos cada vez más complejos para la protección de la información. En este capítulo se abordará la criptología y los diferentes mecanismos y servicios de seguridad como firma y certificados digitales.

En el capítulo 4 se presenta el uso de las bases de datos y datawarehouse enfocándose a estrategias de negocio para las organizaciones, primero se muestran algunos conceptos como tablas, columnas, renglones, etc. En seguida se describe el enfoque relacional ampliamente utilizado y las bases de datos orientadas a objetos, posteriormente se menciona el tema de datawarehouse con todas las herramientas que conlleva como la extracción de información útil mediante la minería de datos.

El capítulo 5 nos presenta diferentes tecnologías para la creación de aplicaciones para organizaciones, ya sea pequeñas o a gran escala, se inicia con UML explicando los beneficios que acarrea, y como junto con RUP hacen una poderosa herramienta de diseño. Asimismo, se describen otras tecnologías para diseñar aplicaciones como RAD y un breve paso por el paradigma orientado a objetos.

En el capítulo 6 se da continuidad al desarrollo de aplicaciones sólo que ahora enfocándose sobre Internet agregando una tecnología más: Multimedia. Se inicia con una revisión del nacimiento de Internet y algunos conceptos inherentes a él, posteriormente se describen algunas tecnologías de desarrollo como HTML, XML, Java y Eclipse. Al final del presente capítulo se revisan conceptos de multimedia y los diferentes elementos que se pueden agregar a los diseños para ser más vistosos e interactivos con los usuarios.

En el capítulo 7 se reúnen todos los elementos para poder enfocar el diseño de aplicaciones al campo de los negocios. En la primera parte se describen conceptos que se refieren al comercio electrónico y en una segunda parte se refieren las fases de la administración de proyectos de tecnologías de información: iniciación, planeación, ejecución y control.

En el capítulo 8 se muestra, como conclusiones, un panorama de lo importante que es estar actualizado en el uso de tecnologías de información, para la oportuna y correcta toma de decisiones en las organizaciones, que al final de cuentas serán usados para fines de negocios.

# Capítulo 1

## Conceptos Básicos y Sistemas Operativos

En el primer capítulo del diplomado se verá el concepto de información profundizando en lo importante que ha sido a través del tiempo adquirirla y en la necesidad de poder transmitirla para las siguientes generaciones, por otra parte se revisará un poco de historia de cómo surgieron las computadoras, qué elementos fueron determinantes para su creación, qué unidades de medición se utilizaron para almacenar información y qué códigos o símbolos se utilizaron para representarla. Paralelamente a la evolución del hardware se desarrollaron los primeros sistemas operativos, revisando de igual manera sus características principales y las desventajas que influyeron en las mejoras continuas hasta tener sistemas operativos modernos.

Para iniciar el presente capítulo se abordarán los conceptos fundamentales que serán base para el desarrollo y mejor comprensión del presente trabajo. Inicialmente se revisará el concepto de información.

### 1.1 Concepto de Información

En sentido general, la información es un conjunto organizado de datos, que constituyen un mensaje sobre un determinado ente o fenómeno. La información se puede abstraer como un fenómeno que proporciona significado o sentido a las cosas e indica mediante códigos y conjunto de datos, los modelos de pensamiento humano. La información, por tanto, procesa y genera el conocimiento humano.

El ser humano a través del tiempo ha tenido la capacidad de generar y perfeccionar tanto códigos, como símbolos con un significado que conforman lenguajes comunes. A partir del establecimiento de sistemas de lenguajes y señales para comunicarse. Se considera a Claude E. Shannon [1] uno de los principales fundadores de lo que se conoce como Teoría de la Información. Asombrosamente los trabajos publicados por Shannon sobre la Teoría de la Información empatan con el mismo principio de entropía que define la medida del desorden de un sistema definido por la física, el cual se abordará en el siguiente punto.

#### Entropía

La física define la entropía como una magnitud termodinámica que mide la energía que no puede utilizarse para producir trabajo. También se puede interpretar como la medida de desorden de un sistema.

La Teoría de la Información define a la entropía como la magnitud que mide la cantidad de información contenida en un flujo de datos, es decir, lo que aporta de información un dato sobre un hecho concreto.

Paralelamente a Shannon, Alan Turing iniciaba los conceptos básicos sobre lo que sentarían las bases de la computadora moderna, su trabajo se le conoce como la Máquina de Turing que a pesar que no pudo plasmar físicamente sus conceptos fueron sumamente importantes para poder desarrollar una computadora, enseguida se describirá lo que Turing imaginaba.

## Máquina de Turing

El antecedente surge con una dicotomía<sup>1</sup> respecto a qué problemas tenían solución y cuáles no. En 1936 Alan Turing se imaginó un dispositivo (una computadora) con capacidad de cálculo y memoria infinita que, si se planteaba correctamente un problema, podía ser capaz de resolver el problema o no resolverlo. Partiendo de lo anterior el concepto aplica de la siguiente manera:

Si un problema era planteado correctamente a esta computadora podía a través de cálculos resolverlo o no resolverlo. Si la computadora paraba (Terminaba sus cálculos) el problema tenía solución; si no paraba significaba 2 cosas: uno, que la solución tiende al infinito o dos que no tiene solución.

Sin duda estos conceptos fueron partida de lo que se avecinaba con el surgimiento de las computadoras modernas. A continuación se revisará un poco de historia, acerca del nacimiento y evolución de las computadoras.

## 1.2 Origen y Evolución de Computadoras

La evolución de las computadoras a través del tiempo ha tenido grandes saltos o características que definen cada periodo, a estos saltos o grandes diferencias se les conoce como generaciones. Algunos autores definen cuatro o cinco generaciones; en el presente trabajo se tomarán como base cuatro generaciones, las cuales se revisarán enseguida.

### Primera Generación (1945 – 1956)

El desarrollo técnico surge con el incremento presupuestal para desarrollar computadoras debido al escenario de la Segunda Guerra Mundial (1939 - 1945).

En 1941, el ingeniero Alemán Konrad Zuse desarrolló una computadora denominada Z3, para diseñar aeroplanos y misiles.

En 1943, los británicos desarrollaron una computadora llamada COLUSSOS, para descifrar códigos secretos e interpretar comunicaciones alemanas. Surgieron computadoras como el Mark1 para calcular trayectorias balísticas para los Estados Unidos; posteriormente el ENIAC que era mil veces más rápida que el Mark1.

En 1945 John Von Newman diseña el EDVAC Computador Automático Electrónico de Variable, aportando características de suma importancia como poseer memoria que permitiera guardar datos.

En 1951 surge la primera computadora comercial llamada UNIVAC1 o Computadora Automática Universal.

Esta primera generación se caracterizó en que el diseño de las computadoras era de propósito específico y tenían una interfaz elemental, difícil de manipular, ya que estaba orientada al hardware. La forma de alimentar instrucciones era vía circuitos alambrados.

### Segunda Generación (1956 – 1963)

Una característica importante en esta Segunda Generación fue la invención del transistor en 1948, éste sustituyó a los grandes tubos de vacío.

---

<sup>1</sup> **Dicotomía.**- División en dos partes

Algunas computadoras creadas bajo esta tecnología fueron el STRECH de IBM y el LARC de Sperry – Rand, dedicadas al ámbito científico.

De igual forma se dieron avances como almacenamiento en disco y memoria, surgen los antecesores de los primeros Sistemas Operativos Modernos, los JCL's (Lenguajes de Control de Tareas).

En esta época surgen lenguajes como Fortran y Cobol. Las instrucciones son alimentadas a las computadoras a través de tarjetas perforadas.

### **Tercera Generación (1964 – 1971)**

En 1958 Jack Kilby, un Ingeniero de Texas Instruments, desarrolló el Circuito Integrado. Esto trajo como consecuencia que las computadoras redujeran su tamaño drásticamente.

Otra característica importante de la época es el surgimiento del concepto de multiprogramación, esto significa que los Sistemas Operativos podían cargar en memoria varios programas simultáneamente.

### **Cuarta Generación (1971 – presente)**

Con la aparición de los Circuitos Integrados, surge la aparición a gran escala. En los 80's se integran cientos de miles de componentes logrando una reducción en el tamaño de las computadoras.

En 1981, IBM introdujo su computadora personal (PC), para huso del hogar, oficina y colegios.

Se dio una expansión en el uso de las computadoras, para 1992 se estaban usando cerca de 65 millones de PC's.

Actualmente la reducción en tamaño ha sido tal que se tienen Laptops, computadoras de billetera o en la palma de la mano (Palms). Por otro lado la interconexión de estos dispositivos ha sido enorme, compartiendo recursos, como memoria, espacio en disco, software, información, etc. Esto hace que se tenga una red Global llamada Internet a lo largo y ancho del Globo terráqueo.

Una persona más que tuvo gran relevancia para el surgimiento de las computadoras fue sin duda el matemático húngaro John von Neumann [2] quien tuvo grandes contribuciones a campos como la física cuántica, la economía, la teoría de conjuntos, entre otros, pero en especial en el campo de la informática, ya que presenta su modelo, el cual lleva su nombre y fue la base para el desarrollo del EDVAC y las presentes computadoras. En seguida se revisará el modelo de Von Newmann.

### **Modelo de Von Newmann**

El matemático húngaro John Von Newmann propone en su modelo [3] (pp. 15) 2 cosas sumamente importantes para la época: 1. Separar el hardware del software y 2. El decir que en un solo dispositivo se podían almacenar tanto instrucciones como datos (memoria). Su modelo consta de Unidades funcionales y en general está dividido en cinco partes (Memoria,



Dispositivos de Entrada / Salida, Unidad Aritmético Lógica (ALU), Unidad de control y bus de datos). En la Figura 1 se muestra en términos generales el modelo propuesto por Newmann.

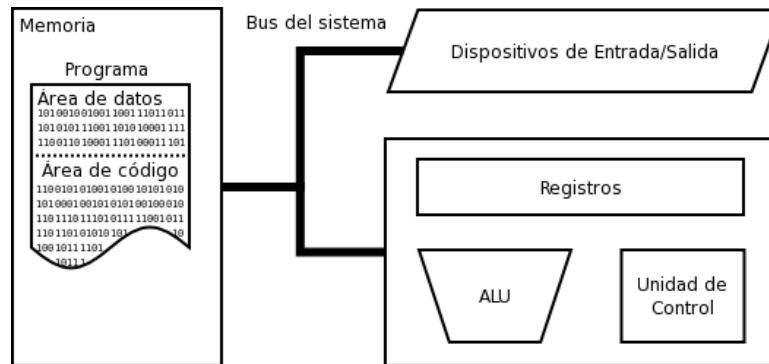


Figura 1.- Modelo de Von Newmann

El modelo de Von Newman fue pionero en el diseño de arquitectura de las computadoras [4] (pp. 7) presentes, en la figura 2 se encuentra el modelo, sólo que ahora representado en bloques funcionales con mayores elementos.

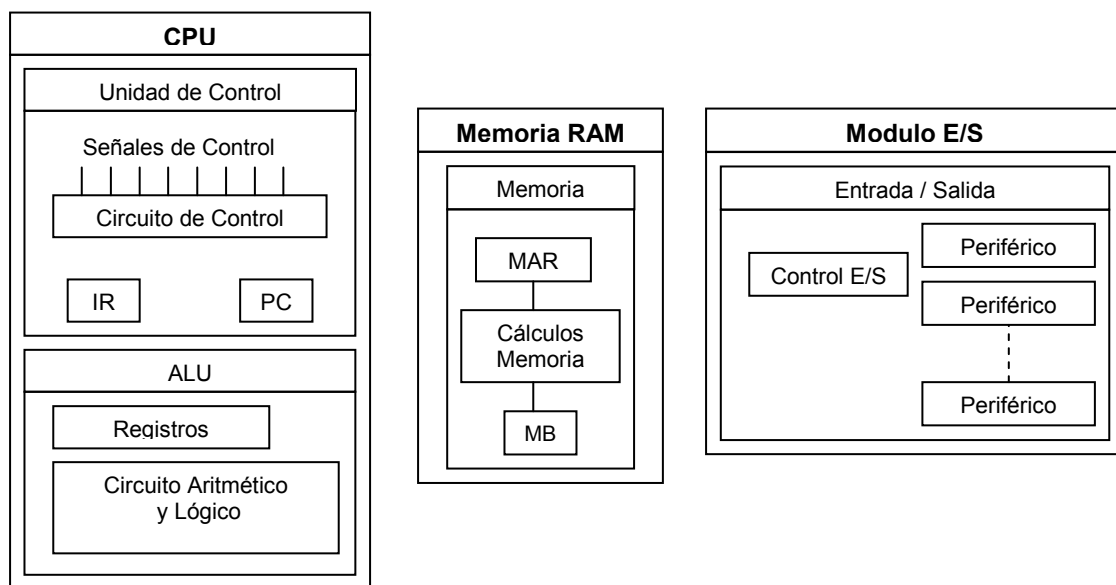


Figura 2.- Bloques Funcionales Von Newman

Los Registros más importantes de la figura 2 son:

1. **PC** El contador de Programa: Contiene la dirección de la siguiente instrucción a ejecutar.
2. **IR** En Registro de Instrucciones se almacena la instrucción que se está ejecutando en cada momento.
3. **Registros ALU**. Los registros para el uso del programador situados en la ALU.
4. **MAR** Registro de Dirección de Memoria. Contiene la dirección de memoria donde va a leer o escribir.
5. **MB** Buffer de Memoria. (También llamado MDR) Contiene la información leída en una operación de lectura o la información a escribirse en una operación de escritura.

La propuesta de Newmann nos indica que un programa consta de una secuencia ordenada de instrucciones. El proceso de ejecución de cada una de estas instrucciones pasa por los siguientes ciclos o fases:

1. **Fase de búsqueda.** Consiste en la lectura de la instrucción y la búsqueda de sus operándos, si los tiene.
2. **Fase de Ejecución.** Consiste en la ejecución de la instrucción.

La fase de la búsqueda consta de las siguientes etapas:

1. Extraer de memoria la instrucción y llevarla al IR, para hacerlo se carga el PC en el MAR y el resultado obtenido se pasa al MB. Después el MB se pasa al IR para que se decodifique dicha instrucción.
2. Incrementar el contenido del PC para que contenga la dirección de la siguiente instrucción.
3. Determinar el tipo de instrucción que se ha extraído y el número de sus operándos.
4. Si la instrucción tiene operándos deberá determinarse su localización.
5. Leer los operándos, si los hay.

La fase de ejecución consta de las siguientes fases:

1. Ejecutar la instrucción.
2. Almacenar los resultados en el lugar apropiado.
3. Volver al paso de búsqueda.

De esta manera se fueron sentando las bases para el nacimiento de las primeras computadoras. Ahora sólo era necesario poder medir la información que se manejaba y almacenaba en las computadoras, es por ello que se verá en el siguiente tema las unidades de información y códigos utilizados para representarla.

### 1.3 Unidades de Información y Códigos

Como sabemos las computadoras utilizan el Sistema Binario, el cual sólo utiliza dos estados: pasa corriente (“on”, “abierto” ó “1”) o no pasa corriente (“off”, “cerrado” ó “0”) y de esta forma puedan manejar y almacenar información.

**Bit y Byte.** Son unidades elementales que representan información en una computadora.

El tamaño, en bits, del byte depende del código que se esté utilizando. Actualmente es más usado el código ASCII extendido, que es un código de 8 bits. Es por eso que un byte tiene 8 bits y cada byte representa un carácter. En la tabla 1 se muestran las unidades de medición [5] en la informática.

Unidad	Abrev	Se habla	Representa	Potencia
1 bit	Bit	Bits	0 ó 1	$2^0$
1 Byte	Byte	Bytes	Conjunto de n bits, n es el tamaño del código.	
1 KiloByte	KB	K'S	1024 Bytes	$2^{10}$
1 MegaByte	MB	Megas	1024 KB	$2^{20}$
1 GigaByte	GB	Gigas	1024 MB	$2^{30}$
1 TeraByte	TB	Teras	1024 GB	$2^{40}$

Tabla 1.- Unidades de Medida

### Códigos de Representación

Inicialmente las computadoras trabajaban con un código ASCII de 7 bits (ISO-646) que podía representar 127 caracteres, posteriormente hubo una extensión a 8 bits llamado ASCII extendido [6] que puede representar 256 caracteres, por otro lado IBM en los grandes sistemas sigue utilizando el código EBCDIC que poco a poco va desapareciendo.

Actualmente se está empezando a trabajar con códigos de 64 bits, el cual es llamado UNICODE y puede representar  $2^{64}$  caracteres. En la tabla 2 se muestra el Código ASCII, en la tabla 3 el ASCII extendido y en la tabla 4 el código EBCDIC.

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

Source: [www.LookupTables.com](http://www.LookupTables.com)

Tabla 2.- Código ASCII (American Standard Code for Information Interchange) (0-127)

128	Ç	144	É	161	í	177	⌘	193	⊥	209	⊖	225	β	241	±
129	ü	145	æ	162	ó	178	⌘	194	⊥	210	⊖	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⊥	211	⊖	227	π	243	≤
131	â	147	ô	164	ñ	180	⊥	196	-	212	⊖	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	⊥	197	⊥	213	⊖	229	σ	245	∫
133	à	149	ò	166	ª	182	⊥	198	⊥	214	⊖	230	μ	246	+
134	â	150	û	167	º	183	⊥	199	⊥	215	⊥	231	τ	247	≈
135	ç	151	ù	168	¿	184	⊥	200	⊥	216	⊥	232	Φ	248	°
136	ê	152	-	169	-	185	⊥	201	⊥	217	⊥	233	Θ	249	.
137	ë	153	Ö	170	¬	186	⊥	202	⊥	218	⊥	234	Ω	250	.
138	è	154	Û	171	½	187	⊥	203	⊥	219	■	235	δ	251	√
139	ì	156	£	172	¼	188	⊥	204	⊥	220	■	236	∞	252	-
140	î	157	¥	173	î	189	⊥	205	=	221	■	237	φ	253	²
141	ï	158	-	174	«	190	⊥	206	⊥	222	■	238	e	254	■
142	Ä	159	f	175	»	191	⊥	207	⊥	223	■	239	∧	255	
143	Å	160	á	176	⌘	192	⊥	208	⊥	224	α	240	=		

Source: [www.LookupTables.com](http://www.LookupTables.com)

Tabla 3.- Código Extendido ASCII

D	H	C	D	H	C	D	H	C	D	H	C
129	81	a	166	A6	w	226	E2	S	78	4E	+
130	82	b	167	A7	x	227	E3	T	79	4F	
131	83	c	168	A8	y	228	E4	U	80	50	&
132	84	d	169	A9	z	229	E5	V	90	5A	!
133	85	e	193	C1	A	230	E6	W	91	5B	\$
134	86	f	194	C2	B	231	E7	X	92	5C	*
135	87	g	195	C3	C	232	E8	Y	93	5D	)
136	88	h	196	C4	D	233	E9	Z	94	5E	;
137	89	i	197	C5	E	240	F0	0	96	60	-
145	91	j	198	C6	F	241	F1	1	97	61	/
146	92	k	199	C7	G	242	F2	2	107	6B	,
147	93	l	200	C8	H	243	F3	3	108	6C	%
148	94	m	201	C9	I	244	F4	4	109	6D	_
149	95	n	209	D1	J	245	F5	5	110	6E	>
150	96	o	210	D2	K	246	F6	6	111	6F	?
151	97	p	211	D3	L	247	F7	7	122	7A	:
152	98	q	212	D4	M	248	F8	8	123	7B	#
153	99	r	213	D5	N	249	F9	9	124	7C	@
162	A2	s	214	D6	O	64	40	Espacio	125	7D	'
163	A3	t	215	D7	P	75	4B	.	126	7E	=
164	A4	u	216	D8	Q	76	4C	<	127	7F	"
165	A5	v	217	D9	R	77	4D	(			

Tabla 4.- Código EBCDIC (Extended Binary-Coded Interchange Code)

### Código UNICODE

Básicamente, las computadoras sólo trabajan con números. Almacenan letras y otros caracteres mediante la asignación de un número a cada uno. Antes de que se inventara Unicode [7], existían cientos de sistemas de codificación distintos para asignar estos números. Ninguna codificación específica podía contener caracteres suficientes: por ejemplo, la Unión Europea, por sí sola, necesita varios sistemas de codificación distintos para cubrir todos sus idiomas. Incluso para un solo idioma como el inglés, no había un único sistema de codificación que se adecuara a todas las letras, signos de puntuación y símbolos técnicos de uso común.

Unicode proporciona un número único para cada carácter, sin importar la plataforma, el programa y el idioma. Líderes de la industria tales como Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Sybase, Unisys y muchos otros han adoptado la norma Unicode. Unicode es un requisito para los estándares modernos tales como XML, Java, ECMAScript (JavaScript), etc. Es compatible con muchos sistemas operativos, con todos los exploradores actuales y con muchos otros productos. La aparición de la norma Unicode y la disponibilidad de herramientas que la respaldan, se encuentran entre las más recientes e importantes tendencias en tecnología de software.

Una vez revisando un poco de historia del surgimiento de las computadoras y el manejo de la información así como su almacenamiento, se revisará lo que define una Arquitectura de computadoras, el cual es el tema siguiente.

## 1.4 Organización, Estructura y Arquitectura de Computadoras

Cuando se habla de arquitectura de computadoras, en realidad se asocian aspectos como; longitud de palabra, de instrucción, conjunto de instrucciones, coprocesadores, modos de

direccionamiento y por otro lado también se refiere a cómo ésta organizada la memoria y cómo se conectan dispositivos de E/S. En el cuadro 1, se muestra en términos generales a qué se refiere cuando se habla de arquitectura de computadoras

Arquitectura de Computadora =  
Conjunto de Instrucciones + Organización de la Máquina.

---

**Cuadro 1.-** Arquitectura de Computadoras

Enseguida se abordará cada concepto que permitirá conocer a qué se refiere y cómo se definen las características de una computadora.

**Longitud de Palabra.** La Longitud de Palabra es referida al Procesador y a la Memoria Principal de la computadora.

Memoria Principal. La longitud de palabra,  $m$ , de la memoria principal, suele coincidir con la del procesador y con el ancho del bus de datos, y es la anchura en bits de una palabra.

Procesador. La longitud de la palabra de datos determina la cantidad de información que es capaz de procesar simultáneamente la unidad central de proceso en cada pulso de reloj. Se mide en bits. A mayor longitud de palabra mayor complejidad y circuitería necesaria en la Unidad Central Procesamiento, pero mayor potencia de proceso.

**Conjunto de Instrucciones.** Un conjunto de instrucciones ó repertorio de instrucciones ó ISA (del inglés instruction set architecture -arquitectura del conjunto de instrucciones-), es una especificación que detalla las instrucciones que una CPU de un ordenador puede entender y ejecutar, o el conjunto de todos los comandos implementados por un diseño particular de una CPU.

**Coprocesador.** La arquitectura de un PC también puede contar con coprocesadores. Estos son microprocesadores especializados en la ejecución de determinados cálculos, que descargan de estas funciones a la unidad central de proceso. Su necesidad depende de la complejidad de las aplicaciones que van a ser ejecutadas.

### **Modos de Direccionamiento**

El modo de direccionamiento especifica una regla para interpretar o modificar el campo de dirección de la instrucción antes de que se haga la referencia real al operando.

Pueden definirse instrucciones con diversos modos de direccionamiento y, en ocasiones, se combinan dos o más modos de direccionamiento en una instrucción.

Sin lugar a dudas el procesador es parte fundamental de la computadora y es por ello que se analizará su estructura y funcionamiento. En la figura 3 se muestra la estructura básica de un procesador mostrando los elementos más importantes:

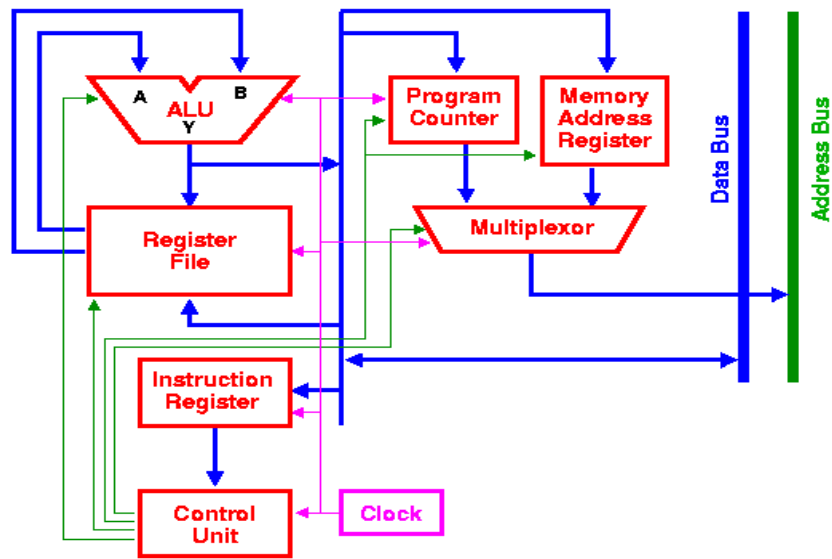


Figura 3.- Estructura Básica de un Procesador

Sus componentes principales son los siguientes:

- **ALU** (Unidad Aritmética y Lógica) La unidad aritmética y lógica, llamada también unidad de cálculo, es la encargada de efectuar el conjunto de operaciones con las que está dotado el sistema digital. Se compone de registros y de un conjunto de circuitos lógicos responsables de realizar las operaciones lógicas y aritméticas prefijadas desde la etapa de diseño del microprocesador.
- **Registro de Archivo** (Register File). Son localidades de almacenamiento (Registros para guardar resultados temporales).
- **Registro de Instrucción** (IR Instruction Register). Se almacena la instrucción que actualmente está siendo ejecutada por el procesador.
- **Unidad de Control** (Control Unit). Se encarga de decodificar la instrucción en el Registro de Instrucción (IR) e inicializa señales que controlan la operación de la mayoría de las otras unidades del procesador.
- **Reloj**. En la actualidad, la inmensa mayoría de procesadores son síncronos, es decir, usan una señal de reloj para determinar cuando capturar la siguiente palabra de datos y realizar una operación sobre ellos.
- **Contador de Programa** (PC). Mantiene la dirección de memoria de la siguiente instrucción a ser ejecutada.
- **Registro de Dirección de Memoria** (MAR). Contiene la dirección de memoria donde se va a leer o escribir.
- **Bus de Dirección** (Adress Bus). Es usado para transferir direcciones a memoria mapeada a los periféricos.
- **Bus de Datos** (Data Bus). Este bus transporta datos desde y hacia el procesador, memoria y periféricos.

Para lograr comunicarse con el procesador fue necesario que surgieran lenguajes que entendieran tanto el programador como el procesador, es por ello que existen diferentes niveles que facilitan dicha tarea. Cada nivel cuenta con un lenguaje que hace la comunicación, en seguida se abordarán estos lenguajes.

## Lenguajes de Programación

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático.

Aunque muchas veces se usa lenguaje de programación y lenguaje informático, como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como, por ejemplo, el HTML.

Se Clasifican en:

1. Lenguaje Máquina
  2. Lenguaje de Bajo Nivel (Ensamblador)
  3. Lenguaje de Alto nivel
1. **Lenguaje Máquina.** Este lenguaje lo entiende fácilmente la computadora, ya que son instrucciones en cadenas de unos y ceros (bits). Para el hombre es sumamente difícil de comprender.
  2. **Lenguaje de Bajo Nivel.** Un lenguaje de bajo nivel está diseñado para simplificar el proceso de proporcionar instrucciones a la computadora evitando las cadenas de bits, por ejemplo el lenguaje ensamblador [8] utiliza mnemotécnicos<sup>2</sup>, siendo más fácil recordarlos que cadenas de unos y ceros.
  3. **Lenguaje de Alto Nivel.** Éste es mucho más fácil de entender y comprender por el ser humano, ya que se utilizan instrucciones con palabras simples parecidas a las que utiliza el ser humano (La mayoría en el lenguaje inglés).

Algunos ejemplos son:

- Lenguaje C
- Pascal
- Java
- Fortran

De igual manera surgieron diferentes sistemas y aplicaciones que podían comunicarse entre los diferentes lenguajes para lograr el entendimiento pleno entre el procesador y el programador. A continuación se estudiarán algunos de los programas que surgieron.

## Traductores de Programas

Un programa traductor tiene la función de convertir código de un lenguaje en otro.

Tipos de Traductores:

- **Ensamblador.** Convierte códigos escritos en lenguaje ensamblador a códigos máquina.
- **Preprocesador.** Traduce un lenguaje de alto nivel a otro, cuando el primero no puede pasar a lenguaje máquina directamente.
- **Interprete.** Realizan un proceso triple (lectura, traducción, ejecución).
- **Compilador.** Traducen código fuente de un lenguaje de alto nivel a código máquina (no siempre)

Para traducir un lenguaje de alto nivel a lenguaje máquina pasan 2 fases:

1. **Análisis.** Se verifica que el programa fuente esté correcto, siguiendo las reglas que define el lenguaje en distintos niveles (léxico, sintáctico, semántico).
2. **Traducción.** Se realiza la traducción, ya sea mediante un compilador o intérprete.

Poco a poco se fue armando el rompecabezas del diseño de una computadora moderna, sólo que falta una pieza fundamental; es el programa que controla y administra las peticiones de usuarios, la correcta comunicación entre los diferentes lenguajes, las peticiones del propio hardware, entre otras cosas. Este programa es el Sistema Operativo, en seguida se revisará su origen y qué características tenían y cómo se fueron desarrollando.

## 1.5 Origen de los Sistemas Operativos

El objetivo que se pretende cubrir en este punto es describir los elementos y necesidades que se dieron para cimentar las bases para el desarrollo de lo que hoy llamamos Sistema Operativo, así como la evolución que ha tenido en el tiempo. En seguida se repasarán algunos puntos dando un paseo a través por la historia, detectando las necesidades que se fueron creando y cómo se fueron resolviendo desde la primera generación de computadoras, hasta lo que hoy se conoce como Sistemas Operativos Modernos.

**Uso Restringido y Desperdicio de Recursos.** La primera necesidad que se tiene con la aparición de las primeras computadoras en la primera generación [1945-1955] es eficientar el uso del procesador y el proceso de alimentación de instrucciones a la máquina, además de evitar desperdiciar recursos.

En la Primera Generación de computadoras la interfaz de comunicación con el usuario era difícil y orientada al hardware. El usuario tenía que tener un completo conocimiento del diseño de la máquina para alambrear su circuito y de esa manera alimentar instrucciones.

Esto, desde luego, tenía como consecuencia que el usuario tenía que estar físicamente con la máquina, por lo cual no cualquier persona estaba capacitada y por otro lado, cuando terminaba y empezaba otro usuario a alambrear su circuito se desperdiciaba mucho el uso del procesador.

**Lenguajes de Alto Nivel.** De lo antes mencionado surge la necesidad de tener una interfaz más fácil entre el usuario y la máquina y viceversa, esto con el fin de que más usuarios pudieran tener acceso a los programas sin estar familiarizados con el diseño.

Esta necesidad se intentó cubrir subiendo el nivel, es decir, crear un lenguaje intermedio que pudiera entender la máquina y el usuario. Desde luego hay que pensar en niveles de traducción en ambos sentidos y es de aquí donde nace el concepto de *máquina multinivel*; de igual manera nace el primer lenguaje de alto nivel llamado FORTRAN, el cual estaba diseñado para calcular numéricamente fórmulas científicas.

**Tarjetas Perforadas.** Otra necesidad que se tiene es eficientar la alimentación de instrucciones a la máquina y esto se resuelve copiando lo que ya se usaba en ese tiempo, la automatización de telares de tejidos en donde se colocaban cartones perforados para que dependiendo de la ausencia de una perforación ó si existía una, significaba un diferente



posicionamiento de las agujas en la tela. Esta misma idea se aplica a las computadoras donde se utilizaron tarjetas perforadas y cada tarjeta generalmente representaba una instrucción en Fortran.

Con esto, ahora el usuario entrega a un operador (humano) su conjunto de tarjetas (Programa en Fortran), éste las colocaba en una máquina lectora y ésta las leía y grababa en una cinta magnética, la cual era la entrada para el compilador de Fortran (que previamente el usuario ya había cargado en memoria), éste lo compilaba y si no había errores se generaba un ejecutable, el cual era ejecutado por el procesador y los resultados eran enviados a una impresora, para finalmente ser entregados al usuario; en caso de que al compilar hubiera errores se mandaban los resultados a la impresora para que de igual forma se le entregaran al usuario, el cual tenía que revisar su programa y repetir todo el proceso.

El proceso de alimentación se mejoró notablemente, ya que esto era mejor que estar alambrando circuitos, sin embargo, todavía hay mucho desperdicio del procesador, cuando el operador toma el siguiente conjunto de tarjetas y las procesa.

**Procesamiento por Lotes.** Para mejorar este proceso la respuesta fue simple, sólo en lugar de colocar un conjunto de tarjetas ahora se colocan un bloque de conjuntos de tarjetas llamados Lotes y así eficientar todo el proceso.

A cada conjunto de programas se le llamó Lote y a la nueva forma de procesarlos se le conoció como procesamiento por lotes.

**Antecedentes de los JCL'S.** Se planteaban las siguientes preguntas:

1. ¿Cómo sabía la computadora dónde terminaba un programa y empezaba otro?
2. Mientras el siguiente programa era cargado a la memoria, ¿Qué hacía el procesador?
3. Cuando se terminaba la ejecución y enviaba los resultados a la impresora, ¿El operador debía esperar hasta que se imprimieran para continuar con el siguiente programa?
4. Si un programa pedía datos, ¿Debía esperar el procesador hasta que los datos fueran enviados?

Las respuestas a estas preguntas nos crean las necesidades para seguir avanzando en el tiempo.

La respuesta a la primera pregunta es que surgen los JCL'S (Lenguajes de Control de Tareas), los cuales eran un software que se encargaba de controlar esas tareas en la máquina.

Para la segunda pregunta, la respuesta es simplemente, nada. Esto da como base que se tenga la necesidad de eficientar los procesos mediante el concepto de multiproceso y multiprogramación.

La tercer y cuarta respuesta es Sí y lo que ayuda a resolver este problema es delegar a los controladores de los dispositivos el trabajo de entrada y salida, con esto se libera al procesador y así podemos seguirlo ocupando.

**Multiproceso y Multiprogramación.** A pesar de todos los avances tan significativos que se dieron todavía, había desperdicio de los recursos de la computadora, como por ejemplo: entre lote y lote el procesador estaba en ocio, o se perdía mucho tiempo cuando el usuario entregaba sus tarjetas, esperaba su listado, corregía sus errores y repetía el proceso. Algo muy importante: en la memoria sólo se podía tener un programa a la vez.

La enorme idea que solucionó varios de estos problemas fue el dividir o segmentar la memoria de la computadora, de tal manera que pudieran ser cargados varios programas al mismo tiempo. A esta implementación se le conoce como Multiproceso, varios procesos que comparten la memoria al mismo tiempo, y multiprograma, varios programas que pueden cargarse en memoria simultáneamente.

Para la correcta implementación surgieron muchos problemas tales como el tamaño de las particiones, el crecimiento de los programas que tienen durante la ejecución, control de direcciones ocupadas y desocupadas, etc.

Desde luego esto fue demasiado para los JCL'S, así que surgen los primeros Sistemas Operativos, los cuales tenían que gestionar los recursos como la memoria y el procesador, además de los dispositivos conectados.

Ahora sólo se tenía que resolver el cómo eficientar en no perder tiempo con traer y llevar tarjetas perforadas.

**Teleproceso.** Para resolver este problema se pensó en intercomunicar dispositivos vía telefónica, lo cual se le conoció con el nombre de Teleproceso. Hay que pensar que en ese tiempo la forma más conocida que había para comunicarse era por medio del teléfono, con esto se reducía mucho el tiempo de trasladarse con tarjetas hasta el lugar físicamente.

Para realizar esta idea se tuvo que cambiar los sistemas de entrada y salida surgiendo el concepto de Terminal, la cual sólo se trata de un Monitor y un Teclado.

Con esto el usuario se conectaba a través de la línea telefónica a la computadora central mediante una Terminal, interactuando directamente como si estuviera físicamente ahí, así a través de un editor le enviaba las instrucciones, la máquina compilaba y le enviaba los resultados a través de la Terminal.

**Tiempo Compartido.** Ahora sólo había que permitir que varios usuarios compartieran los recursos de la computadora. A la facilidad de compartir la computadora con muchos usuarios se le llamó Time Sharing o Tiempo Compartido.

Con esto se dejó atrás a las tarjetas perforadas, lectoras, etc. Y se ganó optimizar el uso de los recursos de la computadora.

Estos puntos fueron sumamente importantes y fueron a través del tiempo construyendo las bases de los primeros sistemas operativos modernos, los puntos antes mencionados se enfocan a las necesidades que se fueron presentando y como se fueron superando con innovaciones tecnológicas, principalmente refiriéndose al hardware. A continuación se estudiarán los primeros Sistemas Operativos enfocándose al software, llevando la misma fórmula, revisando a través del tiempo las necesidades que se crearon y cómo desarrollando innovaciones tecnológicas se fueron superando.

## Primeros Sistemas Operativos

Los primeros Sistemas Operativos [9] eran monstruosamente grandes en su programación, además de no ser modulares dando como consecuencia que cuando se presentaban errores era muy complicado corregirlo. Las grandes innovaciones que se fueron dando para cubrir las necesidades fueron cada vez más complejas al momento de implementarlas.

Un ejemplo de este tipo de sistemas fueron el MULTICS y el THE que surgieron en la segunda generación de las computadoras [1955-1965].

Desde luego que los primeros JCL'S formaron las bases de los sistemas operativos modernos.

Para que se dieran nuevamente aportaciones tecnológicas a la evolución de los Sistemas Operativos se tenían las siguientes necesidades.

1. Extender y ampliar el uso de las computadoras.
2. Facilidad de uso para usuarios menos especializados.
3. Mayor control lógico y automatización.
4. Actualización de hardware.
5. Introducción de nuevos servicios.
6. Tomar ciertas consideraciones de diseño para futuras correcciones del sistema.

**Sistemas por Lotes (Batch).** Surgen por los años 50's y sólo transferían automáticamente el control de un trabajo a otro, utilizaban un software especial llamado monitor, el cual residía en memoria principal.

Un Sistema Operativo por Lotes trabajaba de la siguiente manera:

Los usuarios realizaban sus programas en tarjetas perforadas, las cuales eran entregadas a un operador, éste agrupaba conjuntos de programas con características similares llamados LOTES, estos eran cargados a un dispositivo de entrada para ser ejecutados por el monitor, cada programa, cuando se había procesado y completado, regresaba al estado del monitor el cual podía cargar automáticamente el siguiente programa.

El monitor necesitaba de ciertas características de hardware, además de la lógica de control que poseía como por ejemplo:

- **Protección de memoria.** El usuario no debe alterar el área de memoria que contiene el monitor.
- **Reloj.** Se utiliza un contador para evitar que un simple trabajo monopolice el sistema.
- **Instrucciones privilegiadas.** Ciertas instrucciones sólo deben poder ser ejecutadas por el monitor como las instrucciones de I/O.

**Sistemas Operativos por Lotes Multiprogramados.** Los Sistemas Operativos por lotes solo podían tener cargado un programa a la vez en memoria principal, esto hacía que el procesador permaneciera ocioso y se desperdiciaran recursos mientras se alimentaba con otro. En la Figura 4 podemos observarlo.

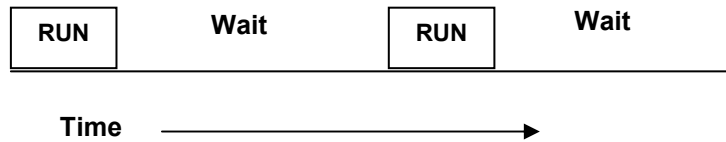


Figura 4.- Sistema Uniprogramado

Con esto se dio la necesidad de evolucionar a la multiprogramación, la cual permitía al CPU ejecutar varios trabajos al mismo tiempo (RUN), así mientras se realizaba una tarea de E/S se cargaba y ejecutaba otro programa. En la Figura 5 nos representa esta evolución.

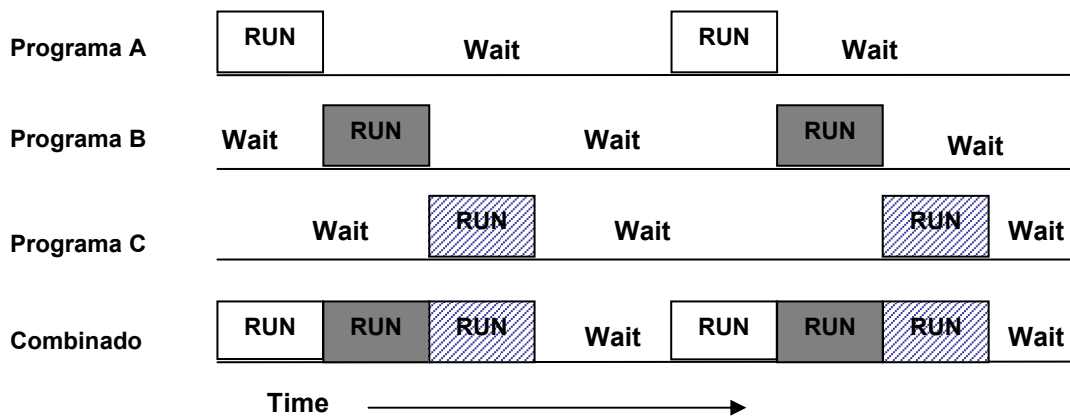


Figura 5.- Multiprogramación

En la figura 5 se observa que se reducen los tiempos de espera del procesador permitiendo aprovechar más sus recursos. La implementación de multiprogramación, desde luego incorpora administración de procesos en la CPU, mecanismos que permitían decidir qué trabajos ejecutar primero, administración de memoria, entre otros.

**Sistemas de Tiempo Compartido (Time Sharing Systems, TSS).** Con la implementación de la multiprogramación surgen los TSS's, los cuales permitían compartir una computadora central con varios usuarios simultáneamente, manejando múltiples trabajos de forma iterativa. Un TSS's brinda porciones de tiempo a cada usuario, eficientando aún más el uso de la CPU.

Para su implementación fue necesario incluir.

- Mecanismos de Administración y Protección de Memoria.
- Manejo de Memoria Virtual
- Sistemas de Archivo y Administración de Discos.
- Administración concurrente de Procesos.
- Mecanismos de Sincronización.

### Tipos de Sistemas Operativos

Se pueden clasificar los sistemas operativos en tres grandes grupos: Sistemas Operativos por su Estructura (Visión interna). Se refiere a cómo está construido internamente, Sistemas Operativos por los Servicios que Ofrecen. En este rubro se puede desmenuzar en varios

subgrupos como nos lo muestra la figura 6, y Sistemas Operativos por la Forma en que Ofrecen sus Servicios (visión externa). Por ejemplo, podemos hablar de sistemas operativos por red o distribuidos.

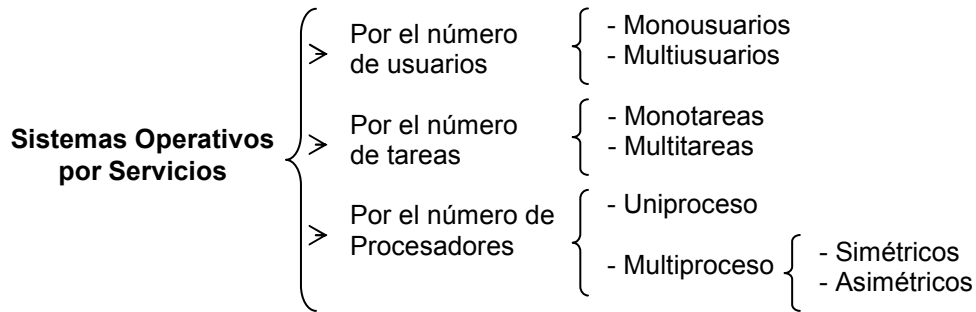


Figura 6.- Sistemas Operativos por Servicios

### Principales Sistemas Operativos

Revisando un poco de historia, en la figura 7 se muestran los principales Sistemas Operativos de 1950 a 2006. Además de mostrar la evolución en el tiempo. Por ejemplo, se observa que Windows nace alrededor de 1985 y su predecesor MS-DOS a principios de la misma década. Una cosa importante que se puede notar es como UNIX ha estado presente desde los 70's hasta la actualidad, siendo este el más potente sistema operativo.

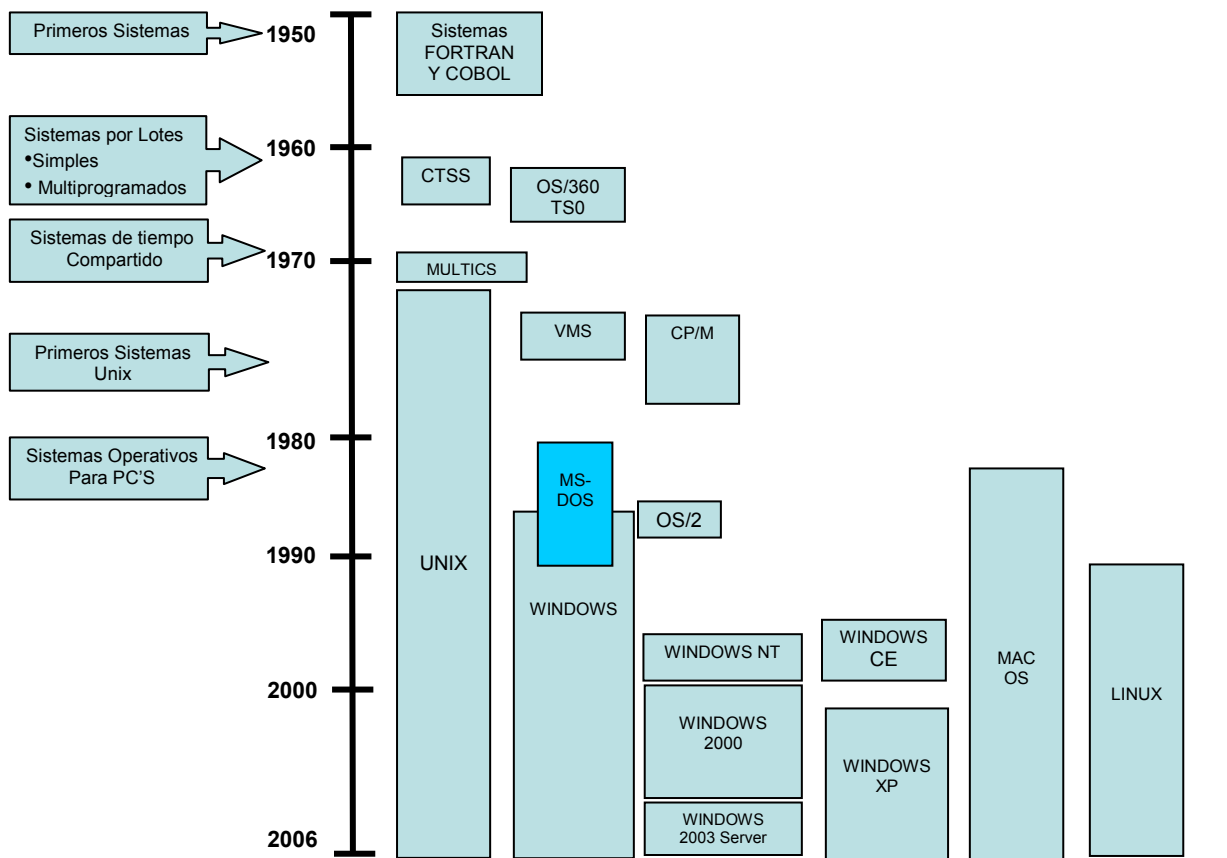


Figura 7.- Sistemas Operativos (1950 – 2006)

Una clasificación más en la que se puede ubicar a los diferentes sistemas operativos es la que se refiere a la forma en que comparten recursos (hardware), enseguida se verá que existen sistemas paralelos, de tiempo real y distribuidos.

**Sistemas Paralelos.** Se les conoce como Sistemas paralelos a los sistemas que cuentan con más de un CPU (Multiprocesador) que comparten el bus de la computadora, reloj y en ocasiones memoria, además de dispositivos periféricos. También se les conoce como Sistemas Fuertemente Acoplados.

Ventajas.

- A mayor número de procesadores, mayor ejecución de trabajos en menor tiempo.
- Tolerancia a Fallas, aumenta estabilidad y confiabilidad del Sistema.
- Ahorro de dinero en comparación con múltiples sistemas con unidades de un solo CPU.

Desventajas.

- Overhead. Es la carga adicional de trabajo generada por el hecho de que varios CPU's trabajen en una misma tarea.

**Sistema de Tiempo Real.** Son sistemas que tienen requerimientos rígidos de tiempo sobre la operación de un CPU.

**Sistemas Distribuidos Débilmente Acoplados.** En una Red de computadoras se manejen diferentes servicios como correo electrónico, FTP o aplicaciones vía HTTP, esto hace que haya varios CPU's ejecutando estas tareas, sin embargo, esta CPU tiene su propia memoria, reloj, etc. A estos sistemas se les conoce como Sistemas Débilmente Acoplados o Sistemas Distribuidos.

Una vez dado un recorrido por la historia del surgimiento de los sistemas operativos se verá enseguida los SO modernos, revisando puntos como estructura, componentes principales y la forma en como operan.

## 1.6 Sistemas Operativos Modernos

Para iniciar se tiene que observar que una característica principal con la que cuentan los Sistemas Operativos Modernos es su diseño estructurado y modular, esto permite independencia de módulos y funciones, lo que facilita enormemente la actualización y corrección de errores.

Un SO es un conjunto complejo de programas que administran y gestionan los recursos del equipo de cómputo, actuando como intermediario entre el usuario y el hardware de la computadora. Tal como se revisó en el anterior punto existe una clasificación de SO por su estructura. Ésta se refiere a cómo ésta construido internamente, por lo que se pueden dividir en dos grupos:

- **Sistemas Monolíticos.** Un Sistema Operativo monolítico está formado por un único proceso compartiendo el mismo espacio de direcciones, las funcionalidades son limitadas por el hardware y no hay niveles de protección.

- **Estructura por Capas.** Este tipo de Sistema divide las tareas en pequeños componentes en lugar de tener un sistema monolítico, y éstos a su vez están interconectados y combinados en un kernel. La principal ventaja es la modularidad que simplifica la depuración y verificación del sistema.

Por otra parte, los SO Modernos podemos desmenuzarlos en sus componentes principales, los cuales se estudiarán a continuación:

Componentes Principales de un Sistema Operativo Moderno:

- Administrador de Procesos
- Administrador de Memoria
- Administrador de Archivos
- Manejador de Dispositivos

Enseguida se describe cuál es la función de cada componente:

**Administrador de Procesos.** Se encarga de Administrar todos los procesos [10] (creación, eliminación, identificación única, seguimiento, contabilidad, asignación y desasignación de recursos).

Es responsable de:

- Crear y eliminar procesos de Usuario y de Sistema
- Suspende y continuar Procesos
- Brindar mecanismos para sincronización de procesos
- Brindar mecanismos para comunicación de procesos
- Brindar mecanismos para el manejo de bloques Mutuos

**Administrador de Memoria.** Se encarga de administrar la memoria de la computadora (Asignar y desasignar celdas, decidir esquemas de cómo auxiliarse de memoria secundaria, controlar celdas libres y ocupadas).

Es responsable de:

- Llevar un registro de las partes de memoria que están siendo utilizadas en el momento y quien los está utilizando.
- Decidir qué procesos se cargarán en memoria cuando haya disponibilidad de espacio.
- Asignar y liberar espacio de la memoria cuando el espacio de la misma esté disponible.

Uno de los principales avances que se dio es el manejo de memoria virtual<sup>3</sup>, el cual surge como necesidad de diseñar esquemas que permitieran incluso duplicar la capacidad física real de memoria.

Memoria virtual es un diseño computacional que permite al software usar más memoria principal (RAM) que la que realmente posee la computadora.

---

<sup>3</sup> **La Memoria Virtual** permite a un software correr en un espacio de memoria que no necesariamente pertenece a la memoria física de una computadora.

Ventajas de la memoria virtual

- Permiten a los programas, direccionar memoria desde un punto de vista lógico, sin tener en cuenta la capacidad física real.
- Cuando un programa se está ejecutando, sólo una porción del programa y los datos pueden actualmente mantenerse en la memoria principal, y la otra parte es colocada en bloques en el disco.

**Administrador de Archivos.** Una característica importante de un Sistema Operativo es poder manipular archivos, teniendo en cuenta que éstos pueden ser almacenados en diferentes dispositivos con características diferentes como velocidad de acceso, capacidad, velocidad de transferencia y métodos de acceso.

Es responsable de:

- Crear y eliminar archivos.
- Crear y eliminar directorios
- Dar soporte a primitivas para manipulación de archivos y directorios.
- Hacer un mapa de los archivos en medios de almacenamiento estables (no volátiles).

**Manejador de Dispositivos.** Su función es controlar y administrar todo dispositivo externo, para eficientar labores como: lectura de datos, salida de datos y comunicación vía controladores.

Este componente presenta al usuario la E/S de datos como una cuestión independiente del dispositivo; es decir, para los usuarios, todos los dispositivos tienen las mismas características y son tratados de la misma forma, siendo el sistema operativo el encargado de atender las particularidades de cada uno de ellos (como su velocidad de operación).

Una técnica muy común, especialmente en salida, es el uso de "*spoolers*". Los datos de salida se almacenan de forma temporal en una cola situada en un dispositivo de almacenamiento masivo (el spool), hasta que el dispositivo periférico requerido se encuentre libre; de este modo se evita que un programa quede retenido porque el periférico no esté disponible. El sistema operativo dispone de llamadas para añadir y eliminar archivos del spool.

Como dispositivos externos, y por tanto también responsable, se encuentran los encargados del almacenamiento secundario como discos o cintas magnéticas.

Su función principal es Administrar los Discos.

Es responsable de:

- Administrar el espacio libre.
- Asignación de almacenamiento.
- Planificación del disco.

Un punto más que es de suma importancia revisar es la manera en como operan los SO Modernos, es por ello que enseguida se profundizará en el tema.



## Operación y Protección

Los SO Modernos pueden administrar los diferentes dispositivos y peticiones de parte del usuario, así como procesos generados por las mismas peticiones a través de interrupciones, protegiendo aspectos como la memoria (áreas reservadas), peticiones de entrada / salida (controlar el tiempo que comparten el CPU sin causar conflictos) y el propio CPU.

Enseguida se escudriñará qué puntos se deben considerar para que un SO pueda operar eficientemente. Primero que nada, en términos generales podemos decir que un SO puede trabajar en Modo Dual.

La mayoría de Sistemas Operativos soporta 2 maneras de ejecución, a esto se le conoce como Modo Dual y el objetivo de éste es proporcionar los medios para proteger al Sistema Operativo de usuarios normales.

Los 2 modos de ejecución son el Modo Usuario y el Modo Supervisor.

**Modo Usuario.** En este modo se ejecutan las aplicaciones del usuario, sólo se ejecutan un subconjunto de instrucciones de todas las soportadas por la arquitectura.

**Modo Supervisor (Privilegiado).** En este modo se ejecutan el Sistema Operativo y el procesador, los cuales al estar en este modo tienen acceso a todo el conjunto de instrucciones soportadas por la arquitectura.

La protección del hardware se logró designando instrucciones privilegiadas, las cuales sólo podrán ser ejecutadas en modo supervisor.

La forma en que opera un SO es impedir que haya colisiones de solicitudes al procesador o que el usuario pueda acceder a partes que sólo el SO debe acceder, es por ello que es parte fundamental proteger aspectos como:

### Protección de E/S

Para evitar que programas de usuario intenten acceder a localidades de memoria asignadas al Sistema Operativo o algún programa se niegue a liberar al CPU, o algún otro conflicto, se implementó lo siguiente:

1. Se definen todas las instrucciones del Sistema Operativo como privilegiadas.
2. Las primitivas de usuario nunca pueden tomar el control de la computadora en modo Supervisor.

### Protección de Memoria

Para proteger el espacio asignado de memoria del Sistema Operativo de los programas de usuario es necesario separarlos y para poder acceder a ellos se utilizan dos registros:

Registro Base. Contiene la dirección física de inicio válida por cada programa.

Registro Límite. Posee el tamaño del rango que indica el límite del bloque de memoria.

## Protección del CPU

El objetivo es impedir que algún programa de usuario se quede en un ciclo infinito no regresando el control al Sistema Operativo.

Para este propósito se utiliza un reloj o Timer, el cual mediante una interrupción se puede regresar el control al Sistema.

Debido a que operaciones de E/S son privilegiadas, sólo pueden ser realizadas por el Sistema Operativo, por lo cual para que un programa de usuario realice una operación de E/S válido o inválido, el programa del usuario debe solicitar al Sistema Operativo que realice estas operaciones por él. A estas solicitudes se les conoce como Llamadas al Sistema.

Los sistemas operativos modernos constantemente están evolucionando y actualmente la manera de operar no es sólo de manera local, sino que deben estar trabajando en red, compartiendo recursos de hardware y software, administrando peticiones de usuarios que se conectan, ya sea desde la misma red local o inclusive desde Internet. Desde luego, las redes fueron un detonante más que hizo que evolucionaran los SO, es por ello que en el siguiente capítulo se profundizará en el tema de las redes y el computo móvil.

## Capítulo 2

### Redes, Telecomunicaciones y Cómputo Móvil

En el siguiente capítulo se verá la forma en que las computadoras se comunican entre sí, cómo la aparición de las redes informáticas trajo enormes beneficios al poder compartir recursos, bajar costos y hacer más rápido procesos sin tener que transportarse físicamente para realizar alguna operación. Sin lugar a dudas las redes se desarrollan no sólo en las computadoras sino ahora en los teléfonos llamados “celulares”, podemos ver como ha incrementado su uso a tal grado que ahora existen más teléfonos móviles que fijos en el mundo. Primeramente se revisarán algunos conceptos básicos repasando un poco de historia para enseguida abordar sus características técnicas para su funcionamiento.

El inicio de las redes de datos surge en la década de los 60's cuando el Departamento de Proyectos e Investigaciones Avanzadas para la Defensa de los Estados Unidos de América (ARPA) iniciaron un proyecto para interconectar computadoras con sus proveedores, posteriormente se fueron sumando más equipos dando como resultado lo que se conoció como ARPANET, precursora de Internet.

Hacia los 70's el crecimiento exponencial de la red evidenció que los protocolos y esquemas de direccionamiento eran poco flexibles. El primero de enero de 1980 se publica el documento conocido como RFC 760, que describe el protocolo de Internet y que dio origen a TCP/IP. Una vez colocado una pequeña introducción estudiaremos algunos conceptos básicos.

#### 2.1 Conceptos, Comunicaciones y Estándares

Protocolo. Un protocolo especifica quién le puede hablar a quién, qué se puede decir y cómo lo pueden decir, la velocidad de comunicación, características eléctricas, el uso de los recursos compartidos y muchas otras variables.

Un *protocolo* es un conjunto de reglas y procedimientos que utilizan dos dispositivos para intercambiar información en una red, sin éstos la comunicación no sería posible.

Pero para que existiera una comunicación entre dos computadoras no bastaba de un protocolo sino de muchos, así que se volvió muy complejo tratar de comunicar dos dispositivos con distintas arquitecturas, es por ello que ISO, en 1977, inicia el desarrollo de OSI, el cual se examinará enseguida.

**Modelo de Referencia OSI.** Debido al crecimiento exponencial de ARPANET en los 70's, se tuvo la necesidad de contar con redes abiertas, ya que no se podían comunicar computadoras de distintos fabricantes.

En 1977, ISO (International Standard Organization) inicia el desarrollo del modelo de referencia llamado OSI [11] (Open System Interconnection), el cual persigue precisamente Interconectar Sistemas Abiertos.

Debido a que la solución no es sencilla, el modelo OSI divide la comunicación entre dos sistemas en 7 capas o niveles, donde cada capa se forma sobre el nivel anterior, quedando como lo muestra la tabla 5:

7	Aplicación
6	Presentación
5	Sesión
4	Transporte
3	Red
2	Enlace de Datos
1	Físico

---

**Tabla 5.-** Modelo de Referencia OSI

Enseguida se describirá la función de cada capa del modelo OSI:

- **Física.** Su objetivo es poner un flujo de bits en un medio físico, expulsándolos del sistema. Tiene que ver con los medios mecánicos, eléctricos y de procedimiento requeridos para la transmisión de datos.
- **Enlace de Datos.** Define el concepto de paquete o trama, define mecanismos para controlar el acceso a la red, control de flujo, detección de errores. Se centra en la interferencia de señales en los medios de transmisión.
- **Red.** La capa de Red proporciona enrutamiento físico de los datos, determinando la ruta entre las máquinas. Examina la topología para determinar la mejor ruta para enviar un mensaje, así como descifrar los sistemas de retransmisión. Regulación de Tráfico, define conceptos de switcheo y Ruteo.
- **Transporte.** Se encarga de garantizar la transmisión fiable de datos a través de una Red. Define la forma en que las conexiones son establecidas y liberadas.
- **Sesión.** En este nivel se lleva un control del estado de la comunicación entre dos nodos.
- **Presentación.** Se encarga de traducir los datos entre distintos formatos y sintaxis usados entre los niveles inferiores y las aplicaciones del nivel superior.
- **Aplicación.** Residen las aplicaciones de red que soportan a las aplicaciones de usuario final.

Por otra parte, existe otro modelo que persigue el mismo objetivo, el cual fue llamado DARPA al inicio y posteriormente TCP/IP.

**Modelo DARPA.** Desarrollado por DARPA (Defense Advance Reserach and Proyects Authority) Departamento de Proyectos e Investigaciones para la Defensa de los Estados Unidos de América y persigue el mismo objetivo que el modelo OSI, sólo que trata de simplificar a 4 capas buscando mejorar el rendimiento. En la Tabla 6 nos muestra una comparación del modelo de referencia OSI y DARPA.



Características de una Red LAN:

- Conexiones Baratas y Sencillas
- Tecnología de transmisión Broadcasting
- Simplicidad
- Confiabilidad
- Transparencia
- Crecimiento sencillo
- Soporte a Múltiples medios de transmisión.

### Medios de transmisión

Una LAN está formada por diferentes equipos conectados de diferentes formas (par trenzado, coaxial, fibra óptica e inalámbricamente) no perdiendo el principio de compartir información y no importando el medio utilizado.



Principales Medios de Transmisión [12] y características

**10Base2.** Llamado Ethernet Delgado (Thin Ethernet), usa un cable coaxial delgado (RG58).

- La longitud de cada segmento de la red no puede superar los 185 m
- Máximo 30 usuarios por segmento
- Usa conectores BNC
- Velocidad en el Bus hasta 10 Mbps
- Impedancia 50ohms

**10Base5.** Llamado Ethernet de cable coaxial grueso (Thick Ethernet).

- La longitud de cada segmento de la red no puede superar los 500 m
- Máximo 100 usuarios por segmento
- Utiliza un conector N sobre el coaxial (Vampiro)
- Velocidad en el Bus hasta 10 Mbps
- Impedancia 50ohms

**10BaseT.** Par Trenzado de Ethernet, puede ser UTP (Unshielded Twisted Pair / Par trenzado sin apantallar), FTP (Foiled Twisted Pair / Par Trenzado con Pantalla Global) o STP (Shielded Twisted Pair ó Par Trenzado Apantallado) 10BaseT solo utiliza los pares dos y tres.

- La longitud de cada segmento de la red no puede superar los 100 m
- Máximo 256 usuarios por segmento
- Usa conectores RJ45 de 8 hilos y RJ11
- Velocidad en el Bus hasta 10 Mbps

Este medio se fabrica de acuerdo a varias categorías que representan el tipo de aplicación para la que están hechos y utilizan un tipo de conector llamado RJ-45 para datos y RJ11 para voz.

**Categoría 1.** Usado en sistemas de telefonía. El nivel de calidad es bueno para voz pero inadecuado para comunicación de datos.

**Categoría 2.** El cable UTP Categoría 2 es empleado para transmisión de voz y datos hasta 4Mbps.

**Categoría 3.** La categoría 3 define los parámetros de transmisión hasta 16 Mbps.

**Categoría 4.** Todos los componentes son probados para un funcionamiento eléctrico de hasta 20Mbps.

**Categoría 5.** Usado para transmisión de data de hasta 100 Mbps.

En la Tabla 7 se presenta en resumen las categorías de 10BaseT, su tipo de cable y la aplicación que tiene cada una de ellas.

Categoría	Tipo de Cable	Aplicación
1	UTP	Voz Analógica
2	UTP	Voz Digital, Datos hasta 1Mbps.
3	UTP, STP, FTP	Datos hasta 16 Mbps
4	UTP, STP, FTP	Datos hasta 20 Mbps
5	UTP, STP, FTP	Datos hasta 100 Mbps
Nivel 6	UTP, STP, FTP	Datos hasta 155 Mbps
Nivel 7	UTP, STP, FTP	Datos hasta 1 Gbps

Tabla 7.- Categorías de 10BaseT

**10BaseFL.** Estándar de cableado para Ethernet que utiliza cable de fibra óptica como medio de transmisión.

- La longitud de cada segmento de la red no puede superar los 2000 m
- Velocidad en el Bus hasta 10 Mbps
- Utiliza conectores ST y SC

Existen 2 tipos de fibra, la multimodo (multimode) o gruesa, es una fibra donde el haz de luz puede ser introducido bajo muchos diferentes ángulos, soporta múltiples lambdas simultáneas y alcanza distancias máximas de 2 Km. La fibra monomodo (Single Mode) soporta una sola transmisión, pero puede alcanzar distancias de hasta 40 Km.

En Resumen los medios de transmisión quedan como lo muestra la tabla 8:

Característica	10Base2	10Base5	10Base T	10Base FL	100Base T
Tasa Datos	10 Mbps	10 Mbps	10 Mbps	10 Mbps	100Mbps
Dis Max Seg	500 m	185 m	100 m	2000 m	100 m
Medios	Coaxial	Coaxial	UTP,STP,FTP	FO	UTP,STP,FTP
Conectores	BNC	BNC	RJ11 y RJ45	SC y ST	RJ11 y RJ45

Tabla 8.- Características medios de transmisión

Hasta este momento se revisaron algunas características de redes de datos cableadas, pero no son el único medio por el cual se puede transmitir, actualmente las redes inalámbricas han tenido un avance significativo, así que se examinará este tema.

**Redes Inalámbricas.** Una red inalámbrica [13] presta esencialmente el mismo servicio que una red cableada tradicional. Sin embargo, la carencia de un cableado hace a la red mucho más flexible: la relocalización de un nodo es inmediata, a diferencia del trabajo que implica mover un nodo en una red convencional. Una red inalámbrica también es una ventaja cuando la disposición física del lugar hace imposible la instalación del cableado.

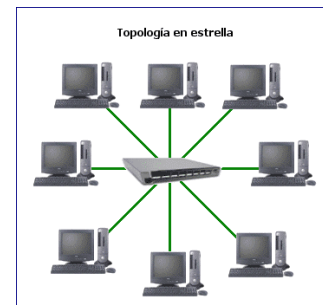
Las redes de datos tal como se revisó, juegan un papel sumamente importante para el desarrollo de las empresas, sin embargo antes de tomar la decisión de implementar alguna, es necesario hacer un estudio previo y saber cuál es la más conveniente, una característica más a tomar en cuenta es la forma en que será conectada (Topología), ya que existen varias formas y cada una de ellas tiene características diferentes. A continuación se verán las topologías de redes de datos existentes y sus características más importantes.

### Topologías de Redes LAN

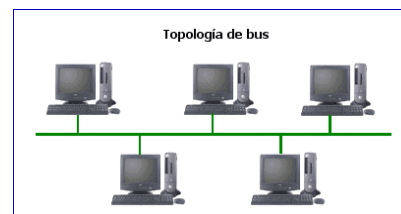
Las redes de computadoras surgieron como una necesidad de interconectar los diferentes host de una empresa o institución para poder así compartir recursos y equipos específicos. Pero los diferentes componentes que van a formar una red se pueden interconectar o unir de diferentes formas, siendo la forma elegida un factor fundamental que va a determinar el rendimiento y la funcionalidad de la red.

La disposición de los diferentes componentes de una red se conoce con el nombre de topología de la red. Las más comunes son las siguientes:

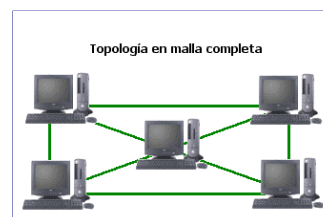
**Topología en Estrella.** Es la más común hoy en día y representa a los nodos unidos punto a punto con un dispositivo central llamado concentrador o HUB.



**Topología Bus Lineal.** Consiste en un solo tramo de cable al que se le conectan los nodos por medio de dos conectores conocidos como conectores "T".

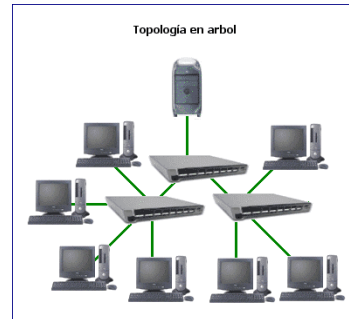


**Topología Malla.** Mezcla Bus Lineal con la estrella, pero con rutas redundantes.

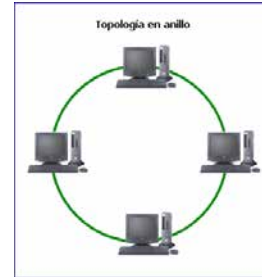




**Topología Árbol.** Mezcla una topología tipo Bus Lineal a la que se le conectan pequeñas estrellas.



**Topología Anillo.** Se basa en Bus Lineal donde los dos extremos, se unen para formar anillo cerrado, teniendo siempre dos rutas posibles.



Desde luego, el estudio de las redes de datos es extenso. Enseguida se realiza una descripción de la manera como viajan los datos por las redes, por supuesto el modelo OSI y el modelo DARPA juegan un papel importante, ya que plantearon las reglas del juego. Lo primero que se examinará es la manera en cómo se encapsulan los datos en una red LAN.

## Encapsulado LAN

Como se revisó anteriormente el Modelo OSI y el modelo DARPA tienen la finalidad de interconectar Sistemas Abiertos y cada modelo divide la comunicación entre dos sistemas en capas o niveles; a su vez cada capa tiene características y maneja protocolos diferentes.

Estos protocolos deben ser transportados entre las capas de más alto nivel sin que estas, deban saber cómo se realiza la comunicación en capas inferiores.

Para llevar a cabo esta tarea se insertan bits al principio de la trama llamados cabeceras (*header*) y al final de la trama llamados cola (*tail*).

A este proceso se le llama encapsulamiento o formato de trama. Los principales servicios de encapsulamiento son:

- **Delimitación.** A cada trama se le agregan marcas de inicio y fin con la finalidad de distinguirse unas de otras,
- **Identificación de Protocolo.** Se incluye un área para identificar los diferentes protocolos de mayor nivel.
- **Direccionamiento.** En todo protocolo de cualquier nivel hay que identificar el origen y el destino.
- **Comprobación de Integridad a nivel de Bit.** Es colocado para la comprobación de que los datos recibidos son idénticos a los que fueron originalmente enviados.

Existen diferentes protocolos que realizan este encapsulamiento, sólo que ahora se describirán los más populares en las redes LAN, por supuesto me refiero a ETHERNET y IEEE802.3.

**Ethernet**

En la década de los 70's en el centro de investigaciones Palo Alto de Xerox se desarrolló un protocolo llamado Ethernet, el cual tiene función en las 2 primeras capas del modelo OSI (Física y Enlace de datos). Existen varias versiones diferentes de Ethernet, cada una con una velocidad de transferencia de datos distinta. La más común es Ethernet II, también llamado 10 Base 5.

En 1979 se estandariza por Xerox Corporation, Intel Corporation y Digital Equipment Corporation.

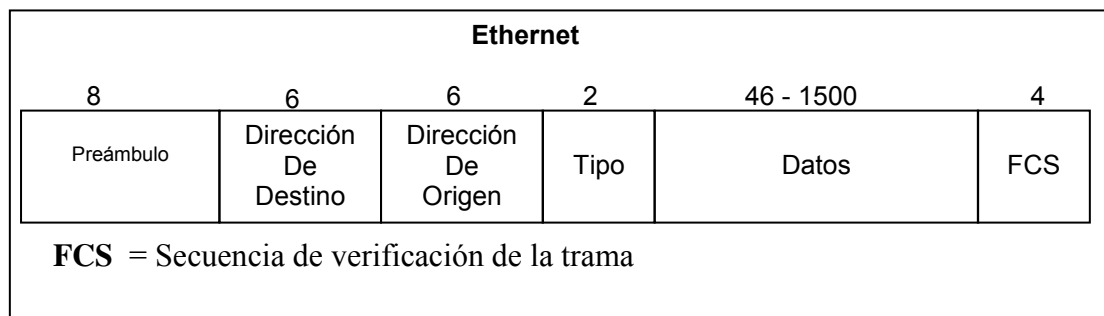
Ethernet depende de un protocolo llamado CSMA/CD (Carrier Sense Multiple Access with Collision Detect) Acceso Múltiple de Detección de Portadora con Detección de Colisiones, en el cual se basa para acceder al medio. Este protocolo se basa en compartir un mismo canal entre varios dispositivos, en donde cada uno compite con los demás, para accederlo, tiene la función de “escuchar” la red para saber si está libre el medio y así poder transmitir.

Las redes Ethernet se consideran redes de difusión, lo que significa que todas las estaciones ven todos los paquetes, sin tener en cuenta si representan un destino determinado. Cada estación debe examinar los paquetes recibidos para determinar si la estación es un destino. En este caso el paquete se pasa a una capa de protocolo superior para su procesamiento adecuado.

Sus principales características de operación son:

- Velocidad: 10 Mbps
- Método de acceso al medio CSMA/CD
- Medios: Par trenzado, coaxial y fibra.

A continuación se describe en la figura 8, cómo está formada la trama Ethernet II y en la parte inferior se describe cada campo.



**Figura 8.-** Trama Ethernet

**Preámbulo.** El preámbulo es una secuencia de bits que se utiliza para sincronizar y estabilizar al medio físico antes de comenzar la transmisión de datos.

**Dirección de destino.** El campo de dirección destino es un campo de 48 bits (6 Bytes) que especifica la dirección MAC de tipo EUI-48 hacia la que se envía la trama

**Dirección de origen.** El campo de la dirección de origen es un campo de 48 bits (6 bytes) que especifica la dirección MAC de tipo EUI-48 desde donde se envía la trama.

**Tipo.** El campo de tipo es un campo de 16 bits (2 bytes) que identifica el protocolo de red de alto nivel asociado con el paquete o en su defecto la longitud del campo de datos. Es interpretado en la capa de enlace de datos.

**Datos.** El campo de datos contiene de 46 a 1500 Bytes. Cada Byte contiene una secuencia arbitraria de valores. El campo de datos es la información recibida del nivel de red.

**FCS.** El campo Secuencia de verificación de la trama (Frame Check Sequence) contiene un valor de verificación CRC (código de redundancia cíclica) de 32 bits o 4 bytes, calculado por el dispositivo emisor en base al contenido de la trama y recalculado por el dispositivo receptor para verificar la integridad de la trama.

Ethernet fue predecesor del siguiente protocolo, el cuál si lo estandarizaron, me refiero a IEEE 802.3 el cual se revisará que solo hay algunas ligeras diferencias.

**IEEE 802.3**

El IEEE (Institute of Electrical and Electronic Engineers) saca una versión compatible con Ethernet a principios de los 80's llamada 802.3, con el esfuerzo de estandarizarlo internacionalmente.

De igual manera este tipo de redes, al igual que Ethernet, se basa en CSMA/CD y son considerados redes de difusión.

Las diferencias con Ethernet son sutiles, a continuación en la figura 9 muestra su trama.

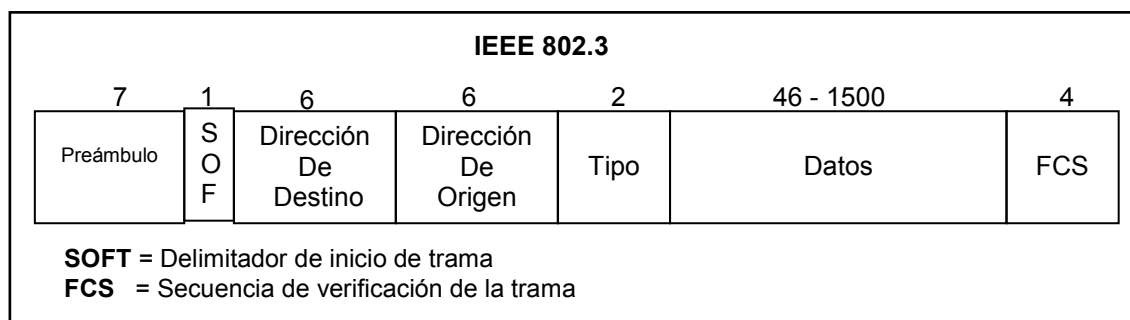



Figura 9.- Trama IEEE802.3

**SOFT.** Delimitación de inicio de trama. Un byte que marca el comienzo de la información propiamente dicho (10101011).

En el cuadro 2 se muestran las especificaciones de IEEE802.3:

	Estándares IEEE 802 para redes LAN
<b>IEEE</b>	Especificaciones de Ethernet (CSMA/CD)
•	802.3u          Fast Ethernet
•	802.3z          Gigabit Ethernet
•	Redes Inalámbricas

---

**Cuadro 2.-** Especificaciones IEEE802.3

Enseguida se describen algunas características de cada especificación IEEE802.3:

**Fast Ethernet.** Debido a la demanda de ancho de banda, en 1995 se publicó el RFC 802.3u conocido como 100 BaseT o Fast Ethernet, tratando de satisfacer los requerimientos de las nuevas aplicaciones.

Características:

- **Full Duplex**

Ethernet nace como un protocolo de medio compartido. Por definición todas las redes compartidas llevan a cabo la comunicación a través del medio half – duplex, lo que significa que una estación transmite mientras las demás deben “escuchar”, esto es, el canal sólo lleva datos en una sola dirección, o transmite o recibe, pero no ambas al mismo tiempo.

Fue necesario que ocurrieran dos eventos importantes para que surgiera la comunicación Full – Duplex:

1. La introducción del cableado 10 BaseT, ofreció la capacidad de separar las trayectorias de transmisión y recepción de datos, lo cual no ofrecía el cable coaxial.
2. La aparición de Ethernet conmutado, trajo como consecuencia que el canal de transmisión ya no se comparte con varios usuarios, sino que permite la conexión entre switches o entre switch y host, en una conexión punto a punto.

- **Utiliza un par de cable UTP para transmisión y otro para recepción.**

Debido a lo anterior, ya no se requiere la función de “escucha a la portadora”, ya que el cable que se utiliza para transmitir y recibir solo con un nodo.

Puesto que las colisiones en medio compartido, ya no se requieren también las funciones de detección de colisiones y los algoritmos de espera.

- **Los datos pueden moverse entre Ethernet y Fast Ethernet sin traducción protocolar.**

Fast Ethernet también usa las mismas aplicaciones y los mismos drivers usados por Ethernet tradicional.

- En muchos casos las instalaciones pueden actualizarse a 100 BaseT sin remplazar el cableado existente.

En el cuadro 3 nos muestra en resumen las características principales de Fast Ethernet.

<p>Características:</p> <ul style="list-style-type: none"><li>• Full Duplex</li><li>• Total compatibilidad con Ethernet</li><li>• Utiliza como método re acceso al medio CSMA/CD</li><li>• Como medio de transmisión soporta par de cobre y fibra óptica (100 Base TX/FX)</li></ul>
---

---

**Cuadro 3.-** Características de Fast Ethernet ó 100 BaseT

**Gigabit Ethernet.** En 1998 se definió para la IEEE el estándar 802.3z o Gigabit Ethernet el cual multiplica por 10 a Fast Ethernet dando como resultado 1000 BaseT. Gigabit Ethernet define para la capa física el uso de la fibra óptica y par de cobre.

**Fibra Óptica multimodo (1000 Base SX y 1000 Base LX).** Un cable de 62.5 micras cubre un máximo de 300 metros (en la especificación SX) o 500 metros (en la especificación LX). Un cable de 50 micras cubre una distancia de 550 metros (para las dos especificaciones SX y LX). La fibra óptica cuenta con ventajas como la inmunidad a interferencias eléctricas, siendo ideal para ambientes propensos a ruidos altos.

Debido a que muchas empresas cuentan con instalaciones de cableado multimodo, éste sea probablemente el estándar a usar.

**Fibra Óptica Monomodo (1000 Base LX).** La máxima distancia que soporta Gigabit Ethernet es de 3000 metros haciendo uso de este tipo de cable. El incremento de la distancia sobre el soporte de la fibra óptica multimodo es la baja dispersión.

En el cuadro 4 se muestran las características principales de Gigabit Ethernet, también llamado 1000 BaseT.

<ul style="list-style-type: none"><li>• Fibra Óptica Multimodo (1000 Base SX y 1000 Base LX)<ul style="list-style-type: none"><li>- Cable de 62.5 micras, 300 Mts SX y 500 Mts LX</li><li>- Cable 50 micras, 550 Mts para SX y LX</li></ul></li><li>• Fibra Óptica Monomodo (1000 Base LX)<ul style="list-style-type: none"><li>- 3000 Metros</li></ul></li><li>• Cable de cobre (1000 Base T)<ul style="list-style-type: none"><li>- Categoría 5</li><li>- 100 metros</li></ul></li></ul>
--

---

**Cuadro 4.-** Características de Gigabit Ethernet ó 1000 BaseT

Una tecnología más que puede transportar información a través de una red es FDDI, el cual utiliza fibra óptica y es utilizado para grandes distancias, se describirá de qué trata esta tecnología y algunas características importantes.

### **FDDI (Fiber Distributed Data Interface) (Interfaz de Datos Distribuida por Fibra)**

La FDDI es una red diseñada para velocidades de 100 Mbps y su principal ventaja es que abarca grandes distancias debido al uso de fibra óptica como medio de transmisión.

FDDI fue desarrollada por ANSI (American Standards Institute) y funciona de manera muy similar al estándar 802.5 (Token Ring), en donde utiliza un método de acceso mediante un token para controlar el tráfico de datos.

Funciona de la siguiente manera:

Si un nodo desea transmitir, toma el control del token<sup>4</sup>, modifica su cabecera e incluye su mensaje o dentro del Token y lo envía al nodo siguiente, quien verifica si es el destinatario, de ser así lee el mensaje y modifica el campo del estado de trama indicando que recibió el mensaje y lo retransmite íntegro al siguiente nodo, el token se va pasando sucesivamente hasta regresar de nuevo al destino, quien vuelve a generar un token de inactividad y lo retransmite a la red.

En cuanto a las redes inalámbricas se desarrollaron diferentes estándares que se describirán enseguida.

### **Redes Inalámbricas 802.11**

Los existentes medios de transmisión como el UTP telefónico y el cable coaxial para televisión siguen siendo considerados dentro de las opciones de conexión de alta velocidad para acceso a redes de datos e Internet, no obstante, ¿Qué pasa con los lugares distantes o que sus características geográficas y topológicas no cuentan con la infraestructura de cableado o no es posible instalarla?. Los sistemas de radio frecuencia contribuyen a desarrollar las comunicaciones en un medio inalámbrico [14].

Las primeras tecnologías inalámbricas operaban a una frecuencia de 900 Mhz y eran de baja velocidad (1 a 2 Mbps). En 1992 se inician las primeras investigaciones y desarrollo de productos operando a 2.4 Ghz, pues en la banda de 900 Mhz se encontraban dificultades con teléfonos celulares e inalámbricos locales.

En Junio de 1997 el IEEE libera el estándar 802.11 para redes de área local inalámbricas. Existen varias versiones para este estándar:

- **802.11.** Trabaja sobre la banda de 2.4 GHz y proporciona anchos de banda de 1 a 2 Mbps, Utiliza dos tecnologías de transmisión.

**FHSS.** Frequency Hope Spread Spectrum, es una tecnología en donde la información transmitida “brinca” de frecuencia cada determinado tiempo en una forma predecible, dentro de un rango finito de frecuencias.

---

**4. Token.-** Es un mensaje especial que circula por la red cuando no hay actividad en el medio.

**DSSS.** Direct Sequence Spread Spectrum. Divide la información en pequeños paquetes que combina antes de enviar con otra secuencia de bits llamada Chpping code, repartiendo cada paquete resultante de diferentes frecuencias del rango.

- **802.11a.** Utiliza la banda de 5 GHz, con un ancho de banda de 54 Mbps, Utiliza como tecnología de transmisión OFDM (Orthogonal Frequency Division Multiplexing). Trabaja dividiendo la señal de radio original de radio en múltiples subseñales que transmite de manera simultánea por diferentes frecuencias.
- **802.11b (WiFi).** Trabaja sobre la banda de los 2.4 GHz, define un ancho de banda de 11 Mbps y utiliza como tecnología de transmisión a DSSS.
- **802.11g** Opera sobre la banda de 2.4 GHz define un ancho de banda de 22 Mbps y utiliza a DSSS como tecnología de transmisión.

Existen otras tecnologías que se desarrollan para enlaces de corta distancia, la más popular y utilizada es Bluetooth.

**Bluetooth.** Esta tecnología crea un concepto llamado PAN (Personal Area Network), con el que fabricantes de equipo electrónico crean dispositivos que se interconectan con computadoras, PDA'S, teléfonos, etc.

Bluetooth opera en la banda de 2.45 GHz y tiene una potencia de 1 Mili Watt, lo cual le da cobertura de alrededor de 10 metros, utiliza a FSHSS como tecnología de transmisión sobre 79 frecuencias.

Las redes de datos se enlazan a través de todo el mundo, a las cuales llamamos WAN, redes de área extensa, y para poder transportar información en un mundo de miles de routers se desarrollan de igual forma protocolos, enseguida se describirán las características y la forma en cómo se desarrolla esta transmisión.

### **Red de Área Extensa (WAN)**

Existen muchas definiciones para una WAN, pero en términos generales podemos decir que una WAN está formada por 2 o más LAN's.

#### Características

- Está formada por 2 más LAN
- Los enlaces suelen ser arrendados por carriers
- Transportan protocolos de nivel local, de una red LAN a otra, por lo que existen diferentes tipos de protocolos y tecnologías.

#### Topologías WAN

- Enlaces punto a punto. Se trata de una conexión dedicada que conecta sólo 2 nodos.
- Enlaces Conmutados. Son conexiones que se establecen y terminan dinámicamente desde un punto a otro en una Red WAN, por ejemplo las líneas telefónicas trabajan con enlaces conmutados.

- Enlaces de Conmutación de Paquetes. Los medios físicos son compartidos entre varios clientes estableciéndose circuitos virtuales o rutas por las que cada paquete de información es llevado de un origen a un destino, en este caso se encuentran ATM, Frame Relay e IP.

## Encapsulado WAN

El término encapsulado describe el proceso de colocar Datagramas IP dentro de un paquete de red para que pueda ser enviado a través de la red subyacente. Existen diferentes protocolos que hacen esto, los más utilizados son los siguientes:

**SLIP.** Data de los principios de los 80's y se diseñó como un método sencillo al solo ofrecer delimitación de tramas.

SLIP presenta algunos problemas:

- No efectúa detección o corrección de errores. Esto lo delega a las capas superiores.
- Sólo reconoce IP. No hay asignación dinámica, esto es muy complicado y costoso brindarle a cada usuario una IP Pública.

**PPP.** En noviembre de 1989 se propuso a PPP como un estándar oficial de Internet. Nace para resolver las deficiencias de SLIP con las siguientes características:

Características:

- Realiza detección de errores.
- Reconoce Múltiples protocolos.
- Permite la negociación de direcciones en el momento de la conexión (Asignación Dinámica).

**X.25.** Sentó las bases de muchos protocolos posteriores y actualmente se sigue utilizando. X.25 se basa en conmutación de paquetes y debido a que fue diseñado sobre medios de transmisión deficientes (ruidosos, interferencias, etc.) incluye varias opciones para garantizar la integridad de la transmisión de información. Es por ello que su característica principal es que es muy confiable aunque es muy lento.

**Frame Relay.** Esta tecnología nace como evolución de X.25 y de igual forma se basa en conmutación de paquetes pero sin tanta sobrecarga en el control y detección de errores, haciendo que sea mucho más rápida la transferencia de información. Frame Relay toma ventaja sobre la evolución de los medios de transmisión como la Fibra Óptica en donde es casi nula la interferencia, disminuyendo los algoritmos de comprobación y detección de errores.

**ATM.** Existen referencias desde los 60's en los Laboratorios Bell; en 1988 la CCITT decide que sería la tecnología del futuro de la red ISDN en banda ancha.



Principales características:

- ATM define tramas fijas a las que llama celdas, las cuales tienen un tamaño de 53 Bytes, 48 de datos y 5 de cabecera.
- Integración fácil de redes clásicas (Ethernet).
- Manejo de ancho de banda bajo demanda.
- Capacidad para manejar la totalidad de tipos de tráfico por la red (voz, imágenes, vídeo, gráficos y multimedia).
- Adaptabilidad tanto en entorno LAN como WAN.
- Manejo de altas velocidades de transmisión (desde 25Mbps hasta 622Mbps)

Hasta este momento, en el presente capítulo se revisaron las redes de datos enfocándose a cómo viaja la información, qué medios de transmisión se utilizan, qué tipos de redes hay, los estándares existentes, las tecnologías que se desarrollan, etc. Ahora se entrará a examinar los elementos que interactúan para que viaje una señal de un punto a otro, es por esto que en el siguiente punto se describirá el tema de telecomunicaciones, mostrando las características principales, ventajas, desventajas de cada elemento.

## 2.3 Telecomunicaciones

En Telecomunicaciones, un sistema de transmisión es un conjunto de elementos interconectados que se utiliza para transmitir una señal [15] de un lugar a otro. La señal transmitida puede ser eléctrica, óptica y radiofrecuencia. Enseguida se describirán algunos sistemas:

**Radiofrecuencia.** Se le conoce así a las ondas aéreas electromagnéticas para comunicar información desde un punto a otro; son portadoras de radio porque desempeñan la función de entregar energía al receptor. Los datos que se transmiten son sobrepuestos sobre la portadora de radio para que pueda extraer de manera precisa por el receptor. Es a lo que se conoce como la modulación de la portadora por la información que se transmite. Después de que los datos son sobrepuestos (modulados) en el transportador de radio, la señal de radio ocupa más de una sola frecuencia, donde la frecuencia de la información modulada se agrega a la portadora. Múltiples portadoras de radio pueden coexistir en el mismo espacio a la vez, sin que haya interferencia, si las ondas de radio se transmiten sobre radiofrecuencias diferentes. Para extraer los datos, un receptor de radio se sintoniza en una radiofrecuencia mientras rechaza otras.

**Fibra Óptica.** La fibra óptica se ha convertido en una de las tecnologías más avanzadas que se utilizan como medio de transmisión. Este novedoso material vino a revolucionar los procesos de las telecomunicaciones en todos los sentidos, desde lograr una mayor velocidad y disminuir casi en su totalidad los ruidos y las interferencias hasta multiplicar las formas de envío en comunicaciones y recepción por vía telefónica.

La fibra óptica tiene muchas ventajas:

1. Gran flexibilidad, el radio de curvatura puede ser inferior a 1 cm, lo que facilita la instalación enormemente
2. Gran ligereza, el peso es del orden de algunos gramos por kilómetro, lo que resulta unas nueve veces menos que el de un cable convencional

3. Facilidad para localizar los cortes gracias a un proceso basado en la telemetría, lo que permite detectar rápidamente el lugar y posterior reparación de la avería, simplificando la labor de mantenimiento.

Sin embargo presenta algunos inconvenientes:

- equipos terminales aún demasiado costosos
- la especialización del personal encargado de realizar las soldaduras y empalmes.

**Satélites.** La comunicación Satelital funciona básicamente igual a la comunicación de radio, sólo que se utilizan a los satélites como repetidores.

Ventajas:

- Rápida Instalación, sólo necesitan montarse y orientarse al satélite para empezar a transmitir.
- Adaptable a cualquier tipo de terreno.
- Permite movilidad.
- Generalmente sus huellas cubren varios países.

Desventajas:

- Requiere permisos para el uso de frecuencias.
- Susceptible a interferencias principalmente climáticas.
- Los satélites tienen un tiempo de vida limitado.
- El ancho de banda es muy caro.

**Cables de cobre.** Existe una diversidad de cable para comunicaciones pero en general tenemos el cable coaxial y el par de cobre en sus diferentes categorías.

Ventajas:

- Cuenta con una mayor base instalada de todos los medios de transmisión.
- Llegan al usuario final.

Desventajas:

- El ancho de banda es menor al de una fibra óptica
- Es susceptible a interferencias electromagnéticas.
- Es propenso a deteriorarse por condiciones ambientales.

Siguiendo con lo técnico, se irá desmenuzando cada vez más las redes de datos y todo lo que implica transportar información, se verá a continuación un protocolo, que en realidad es un conjunto de protocolos TCP/IP, con el que gracias a él podemos entrar a la red de redes, "Internet".

## 2.4 TCP/IP

En 1973, la Agencia de Proyectos de Investigación Avanzada (DARPA), para la defensa de los Estados Unidos, inició un programa para la investigación de tecnologías que permitieran la transmisión de paquetes de información entre redes de diferentes tipos y características. El proyecto tenía por objetivo la interconexión de redes, por lo que se denominó "Internetting", y a la familia de redes de computadoras que surgieron de esta investigación se les denominó "Internet". Los protocolos desarrollados se denominaron como el Conjunto de Protocolos TCP/IP [16], que surgieron de dos conjuntos desarrollados; los protocolos de control de transmisión (Transmission Control Protocol) e Internet (Internet Protocol).

Aunque es común utilizar a TCP /IP como si fuera un solo protocolo en realidad son varios, de entre los cuales destaca el protocolo IP. Bajo este nombre (TCP/IP) se esconde uno de los protocolos más usados del mundo, debido a que es el más usado por Internet y esta muy extendido en el sistema operativo Unix, entre otros sistemas operativos como: Linux, Windows NT, Novell, etc.

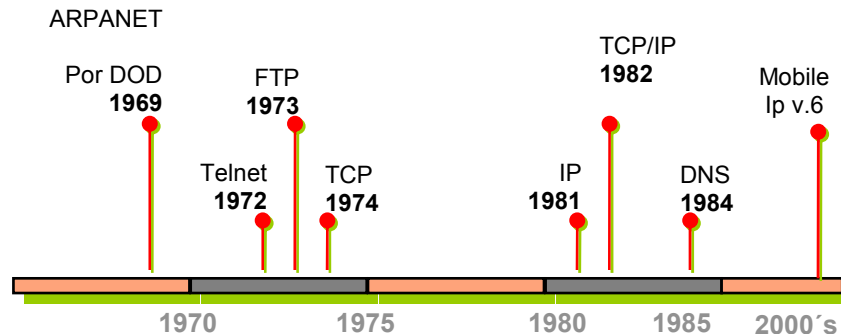


Figura 10.- Nacimiento de TCP/IP

En la figura 10 se puede observar la ubicación del nacimiento de TCP/IP en una línea de tiempo.

### Características de TCP/IP

- Suite de Protocolos Estándares.
- Ruteable (IP).
- Tecnología de conexión de sistemas diferentes.
- Robusto, escalable, trabaja en ambiente Cliente-Servidor.
- Protocolo de Internet.

La Arquitectura de Internet está basada en capas. Esto hace más fácil implementar nuevos protocolos. El conjunto de protocolos TCP/IP, al estar integrado plenamente en Internet, también dispone de este tipo de arquitectura. El modelo de capas de TCP/IP es algo diferente al propuesto por ISO (Internacional Standard Organization) para la interconexión de sistemas abiertos (OSI). En la figura 11 se muestra una comparativa entre el modelo de referencia OSI y el modelo TCP/IP.

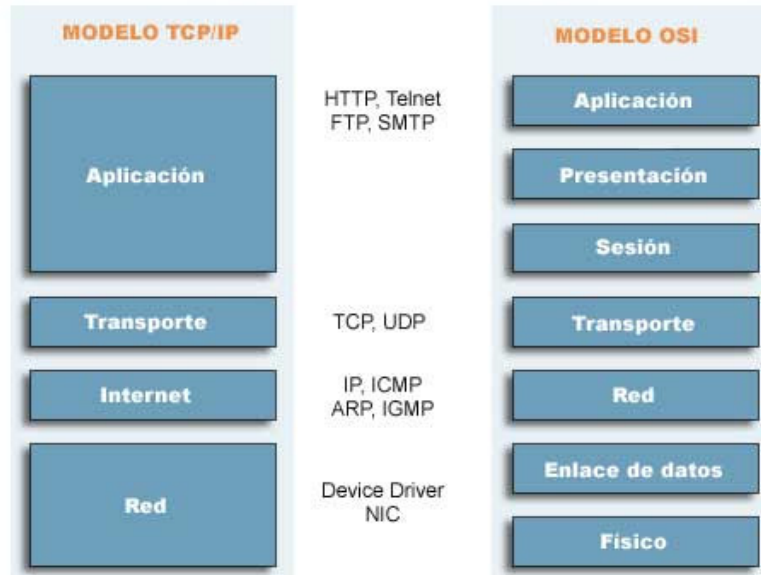


Figura 11.- Correspondencia capas modelos TCP/IP y OSI

La arquitectura basada en TCP/IP propone 4 capas en las que las funciones de las capas de Sesión y Presentación son responsabilidad de la capa de Aplicación, y las capas de Enlace de Datos y Física son vistas por la capa de Interfaz de Red.

Como puede verse TCP/IP presupone independencia del medio físico de comunicación, sin embargo, existen estándares bien definidos en la capa de Enlace de Datos y Física que proveen mecanismos de acceso a los diferentes medios y que en el modelo TCP/IP deben considerarse la capa de interfaz de Red; siendo los más usuales el proyecto IEEE 802: Ethernet, Token Ring y FDDI.

En la figura 12 se observan las capas de TCP/IP mostrando características como protocolos, tecnologías utilizadas, etc.

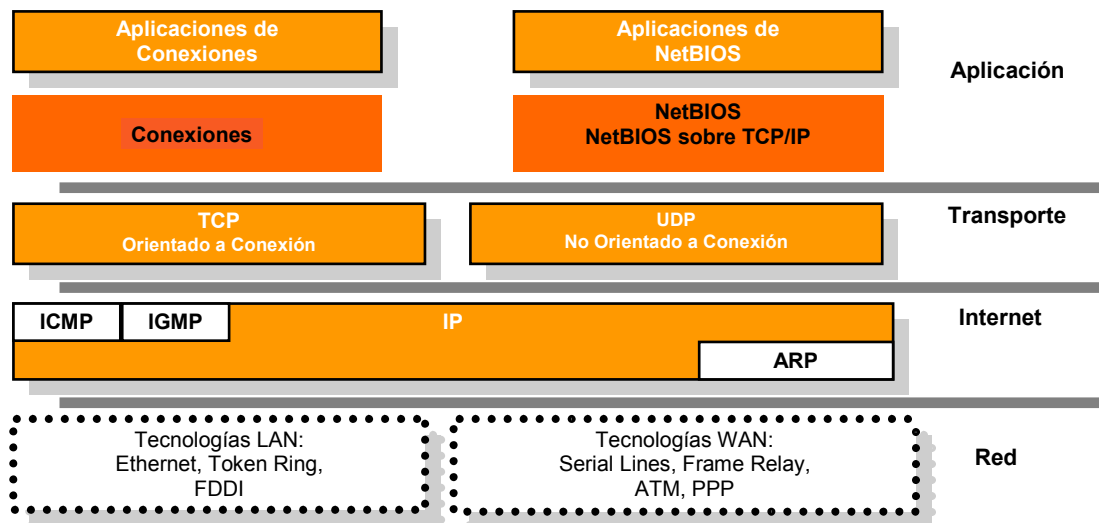


Figura 12.- Algunas características de las capas de TCP/IP

En la capa 2 llamada “Internet” por el modelo TCP/IP se observa que es muy importante el protocolo IP del cual se describirá a continuación:

**Protocolo de Internet (IP).** El protocolo IP es un protocolo que trabaja a través de una sola dirección IP, para cada host, donde cada host puede ser: una impresora, una computadora, un ruteador, un servidor, etc. Cada uno de ellos tiene una sola dirección IP, para que de esta manera puedan compartir o conectarse a los recursos de la red, en caso de existir más de dos dispositivos con una misma dirección IP, se dice que entran en conflicto y no es posible realizar la conexión.

Características:

- Direcciona y reenvía paquetes.
- No orientado a conexión.
- Transmisiones en unidades denominadas Datagramas.
- Sin corrección de errores, ni control de congestión.
- No garantiza la entrega en secuencia.

El protocolo IP trabaja a través de direcciones IP, ¿Y qué es una dirección IP?, bueno pues enseguida se entrará al tema describiendo algunas cosas importantes para la mejor comprensión del tema.

## Dirección IP

La Dirección IP es una dirección de 32 bits dividida en 4 octetos que sirve para identificar a un equipo o dispositivo que esta trabajando con TCP/IP, esta dirección debe ser única y no puede estar repetida.

Conceptualmente, cada Dirección IP está compuesta por un par de **Net ID** y **Host ID**, en donde se identifica la Red y el Host dentro de la Red.

En la figura 13 se muestra cómo se divide la dirección IP en un NET ID y un HOST ID.

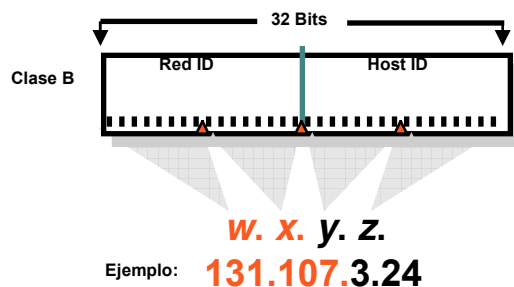


Figura 13.- Net ID y Host ID en una dirección IP.

Una dirección IP se clasifica por clases, para una administración de las mismas. La clase se identifica mediante las primeras secuencias de bits. Existen cinco clases de direcciones IP, para saber a qué clase de dirección pertenece una dirección IP, se tiene que observar el primer octeto de la izquierda a derecha (de orden superior). Aunque existe en este octeto la posibilidad de tener 256 direcciones, por cuestiones operativas sólo se utilizan 254.

En la tabla 9 se muestran las clases de IP que se manejan mostrando los rangos que utilizan y cantidad de nodos por cada clase.

Clases Direcciones IP	Formato	Rango de direcciones	Máximo de hosts
A	R.H.H.H	1.0.0.0 a 126.0.0.0	$2^{24}-2$
B	R.R.H.H	128.1.0.0 a 191.254.0.0	$2^{16}-2$
C	R.R.R.H	192.0.1.0 a 223.255.254.0	$2^8-2$
D	No utilizado	224.0.0.0 a 239.255.255.255	No utilizados
E	No utilizado	240.0.0.0 a 254.255.255.255	No utilizados

Tabla 9.- Clases IP

Las direcciones IP son la base fundamental para la existencia en una red, ya sea local o de área amplia, e intervienen con los diferentes medios de transmisión existentes, cableado o inalámbrico. En los siguientes puntos se detallarán algunos aspectos técnicos para la existencia de las redes inalámbricas.

## 2.5 Redes Inalámbricas

En los últimos años las redes de área local inalámbricas (WLAN, Wireless Local Area Network) están ganando mucha popularidad, que se ve acrecentada conforme sus prestaciones aumentan y se descubren nuevas aplicaciones para ellas.

La norma IEEE 802.11 fue diseñada para sustituir a la capa Física y MAC de la norma 802.3 (Ethernet), así, la única diferencia entre ambas es la manera en la que los dispositivos acceden a la red, por lo que ambas normas son perfectamente compatibles.

En el caso de las redes locales inalámbricas, está clara la cada vez mayor imposición del sistema normalizado por IEEE con el nombre 802.11g, norma conocida como Wi-Fi o Wireless Fidelity, aprobada en 1990 y basada en el modelo OSI (Open System Interconnection), la primera norma 802.11 utilizaba infrarrojos como medio de transmisión para pasar hoy en día al uso de radiofrecuencia en la banda de 2.4 Ghz, con este sistema podemos establecer redes a velocidades que pueden alcanzar desde los 11 Mbps hasta los 54 Mbps estándares en los equipos actuales, aunque es posible alcanzar mayores velocidades.

Ventajas:

- **Movilidad:** Las redes inalámbricas proporcionan a los usuarios de una LAN acceso a la información en tiempo real en cualquier lugar dentro de la organización o el entorno público (zona limitada) en el que están desplegadas.
- **Simplicidad y rapidez en la instalación:** La instalación de una WLAN es rápida y fácil y elimina la necesidad de tirar cables a través de paredes y techos.
- **Flexibilidad en la instalación:** La tecnología inalámbrica permite a la red llegar a puntos de difícil acceso para una LAN cableada.

- **Costo de propiedad reducido:** Mientras que la inversión inicial requerida para una red inalámbrica puede ser más alta que el costo en hardware de una LAN, la inversión de toda la instalación y el costo durante el ciclo de vida puede ser significativamente inferior.
- **Escalabilidad:** los sistemas de WLAN pueden ser configurados en una variedad de topologías para satisfacer las necesidades de las instalaciones y aplicaciones específicas. Las configuraciones son muy fáciles de cambiar y además resulta muy fácil la incorporación de nuevos usuarios a la red.

Desventajas:

- **Elevado costo inicial:** Pues los dispositivos inalámbricos cuestan hasta tres veces más que uno para cables.
- **Baja velocidad de transferencia de datos:** En comparación con las redes con cables, que alcanzan velocidades mucho mayores.
- **Seguridad:** Debido a que la información viaja por el aire es sumamente vulnerable a ataques.

Las redes inalámbricas no se han desarrollado tan rápidamente como de manera cableada, sin embargo, está ganando mucho terreno, existen necesidades que poco a poco se han superado, los problemas fundamentales son técnicos y para poder comprender esto enseguida se examinará cómo viaja la información en forma inalámbrica, iniciando con algunos conceptos básicos de radiofrecuencia.

## 2.6 Fundamentos de Radiofrecuencia

El término radiofrecuencia, también denominado *espectro de radiofrecuencia* o *RF*, se aplica a la porción del espectro electromagnético en el que se pueden generar ondas electromagnéticas aplicando corriente alterna a una antena. Las ondas electromagnéticas están presentes en todas las frecuencias, sin embargo, sólo se puede disponer de una parte del espectro para transmitir señales; este subconjunto es referido como Espectro de Radio Frecuencia (RF) y los rangos van desde los 9 Khz hasta los 300 Ghz.

### Propiedades de una señal

Para entender cómo transmitir datos vía inalámbrica es necesario tener claro algunos conceptos sobre las ondas. La onda senoidal es la más simple y común de ellas pero de igual forma muy práctica para ver algunas propiedades que aplican a cualquier tipo de onda. Para entender mejor esto, la figura 14 muestra sus características:

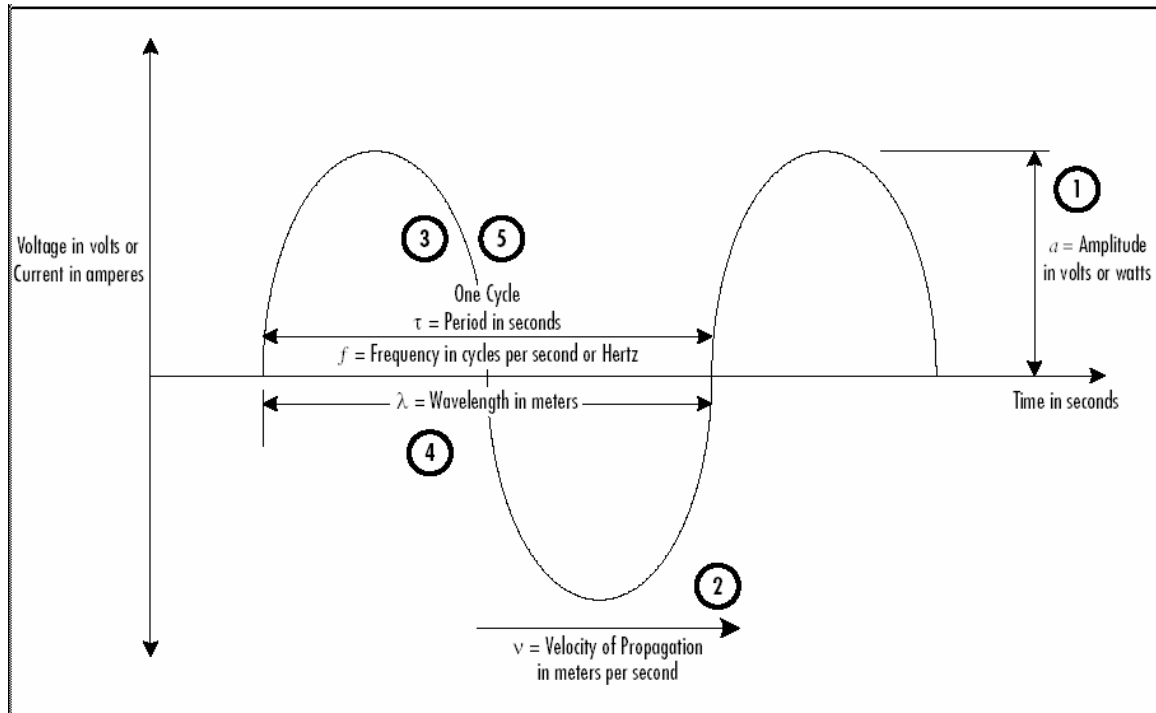


Figura 14.- Características de la Onda Senoidal

1. Es la amplitud en voltios o amperios.
2. Velocidad de Propagación. Se refiere a la velocidad que la onda viaja en un medio, sus unidades son en metros por segundo.
3. Periodo (T). El tiempo que transcurre en un ciclo y su unidad es el segundo.
4. Longitud de onda ( $\lambda$ ). La distancia que la onda se propaga en un ciclo, su unidad es el metro.
5. Frecuencia (f). Es el número de ciclos que repite la onda por segundo. Su unidad son los Hertz.

¿El porqué es importante estudiar las ondas?, bueno la información viaja en ellas cuando se transporta información de un punto a otro en forma inalámbrica. Existen diferentes técnicas para poder montar la información en las ondas electromagnéticas, en el siguiente punto se revisarán estas técnicas.

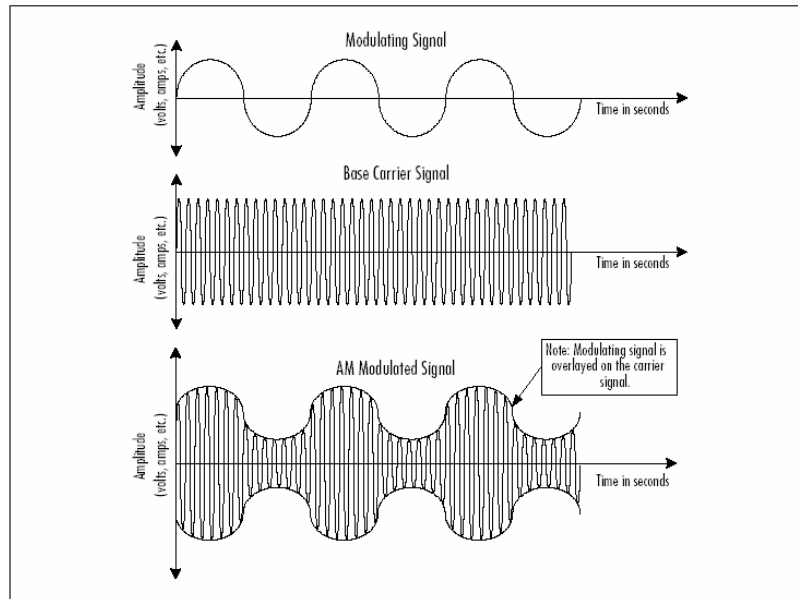
Técnicas de Modulación [17]. El término modulación engloba el conjunto de técnicas para transportar información sobre una onda portadora, típicamente una onda senoidal. Estas técnicas modifican las propiedades de la señal. Existen diferentes técnicas para modificar las propiedades dependiendo de que tipo de información se vaya a transferir, digital o analógica.

Esquemas de modulación Analógica. Existen dos esquemas de modulación analógica que son ampliamente utilizados:



**Amplitud Modulada**

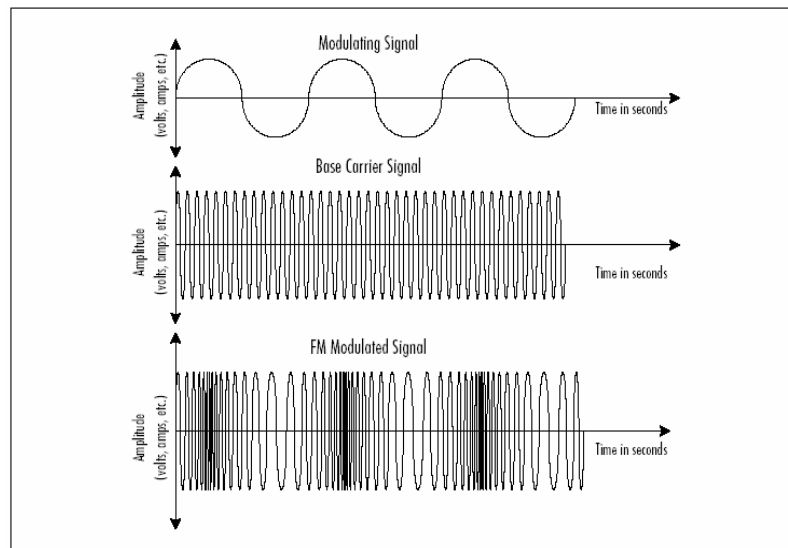
Modifica la amplitud de una onda portadora como lo muestra la figura 15.



**Figura 15.- Amplitud Modulada**

**Frecuencia Modulada**

En este caso modifica la frecuencia de la señal portadora, como lo muestra la figura 16.



**Figura 16.- Frecuencia Modulada**

Esquemas de Modulación Digital. Con el uso de la computadora es necesario transformar o codificar señales analógicas en binarias o digitales. A este hecho se le conoce como digitalización.

Los esquemas más simples de modulación digital son:

1. On/Off Keying (OOK)
2. Frecuency Shift Keying (FSK)
3. Pulse Amplitude Modulation (PAM)
4. Phase Shift Keying (PSK)

**On/Off Keying (OOK)**

Es una técnica de las más simples, el concepto es si transmite representa un "1" y si no transmite representa un "0". Se puede observar en la figura 17.

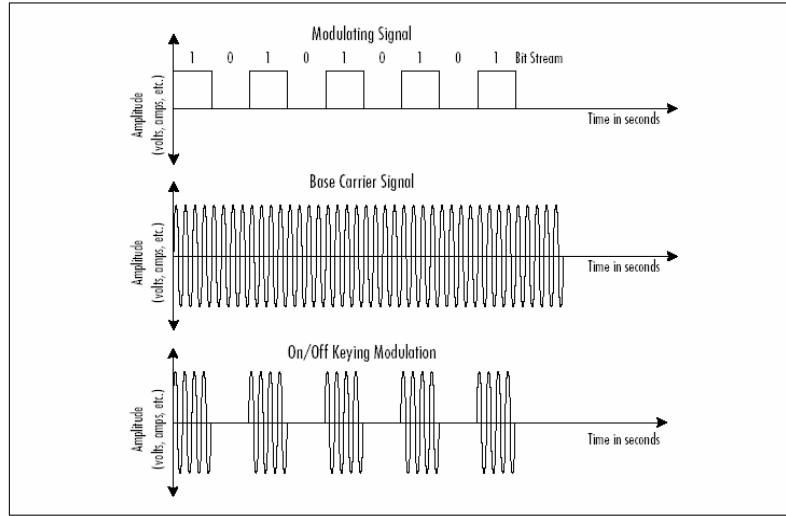


Figura 17.- On/Off Keying (OOK)

**Frequency Shift Keying (FSK)**

Para representar los dos estados necesarios modifica la frecuencia en la portadora. Representa un "1" cuando se aumenta la frecuencia y representa un cero "0" cuando la emite en forma normal. Se puede observar en la figura 18.

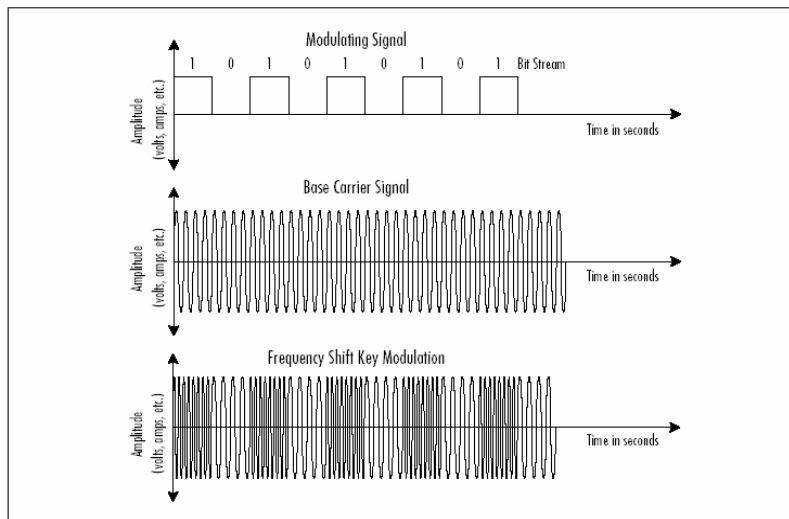


Figura 18.- Frecuency Shift Keyeng (FSK)

**Pulse Amplitude Modulation (PAM)**

Este esquema de modulación modifica la amplitud de la portadora como lo muestra la figura 19.

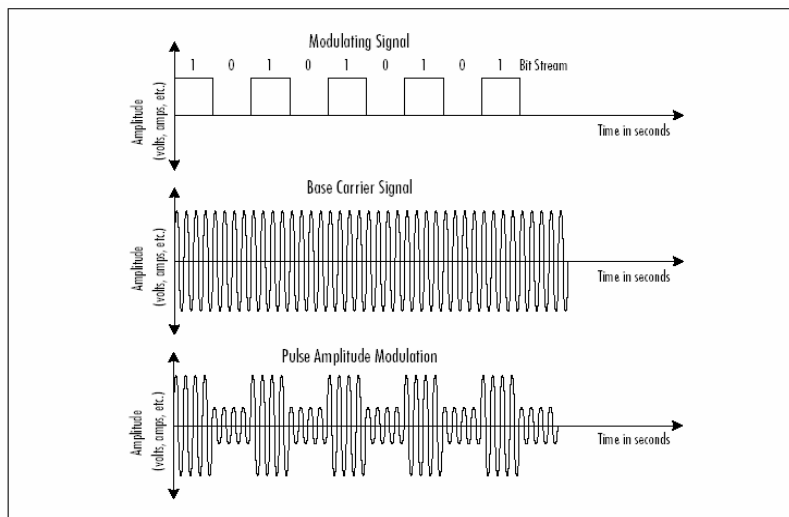


Figura 19.- Pulse Amplitude Modulation (PAM)

### Phase Shift Keying(PSK)

Este método es uno de los más utilizados en las comunicaciones, modificando la fase de la señal portadora. En la figura 20 se muestra este método.

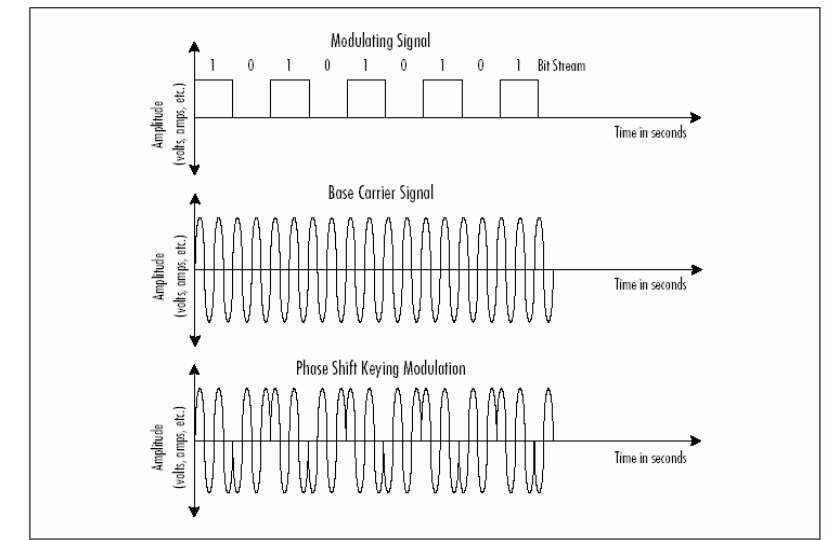


Figura 20.- Phase Shift Keying (PSK)

Para que la información pueda viajar a través de las ondas se han desarrollado diferentes tecnologías, enseguida se describirán las más importantes:

## 2.7 Tecnologías de Intercomunicación

Una tecnología de Interconexión tiene como objetivo transportar información de un punto a otro de manera inalámbrica, las tecnologías más utilizadas son las siguientes:

**Microondas.** Son ondas electromagnéticas de radio situadas en el espectro electromagnético entre los rayos infrarrojos (cuya frecuencia es mayor) y las ondas de radio. Su longitud de onda va aproximadamente desde 1 mm hasta 30 cm. Las microondas se generan con tubos de electrones especiales como el clistrón o el magnetrón, que incorporan resonadores para controlar la frecuencia, o con osciladores o dispositivos de estado sólido especiales. Las microondas tienen muchas aplicaciones: radio y televisión, radares, meteorología, comunicaciones vía satélite, medición de distancias, investigación de las propiedades de la materia o cocinado de alimentos.

La exposición a las microondas es peligrosa cuando se producen densidades elevadas de radiación. Pueden provocar quemaduras, cataratas, daños en el sistema nervioso y esterilidad. Todavía no se conocen bien los posibles peligros de la exposición prolongada a microondas de bajo nivel.

**Rayos Infrarrojos.** Emisión de energía en forma de ondas electromagnéticas en la zona del espectro situada inmediatamente después de la zona roja de la radiación visible. La longitud de onda de los rayos infrarrojos es menor que la de las ondas de radio y mayor que la de la luz visible. Oscila entre aproximadamente  $10^{-6}$  y  $10^{-3}$  metros. La radiación infrarroja puede detectarse como calor.

Los rayos infrarrojos se utilizan para obtener imágenes de objetos lejanos ocultos por la bruma atmosférica, que dispersa la luz visible pero no la radiación infrarroja. En astronomía se utilizan los rayos infrarrojos para estudiar determinadas estrellas y nebulosas.

**Espectro Amplio.** La gran mayoría de los sistemas inalámbricos emplean la tecnología de Espectro Extendido (Spread Spectrum), una tecnología de banda amplia desarrollada por los militares estadounidenses que provee comunicaciones seguras, confiables y de misión crítica. La tecnología de Espectro Amplio está diseñada para intercambiar eficiencia en ancho de banda por confiabilidad, integridad y seguridad.

Existen dos tipos de señales de Espectro Amplio: Salto en Frecuencia (Frequency Hopping, FH) y Secuencia Directa (Direct Sequence, DS).

- Espectro Amplio con salto en frecuencia (FHSS)

FHSS utiliza una portadora que cambia la frecuencia en un patrón conocido tanto por el transmisor como por el receptor. Tanto transmisor como el receptor están debidamente sincronizados comunicándose por un canal que está cambiado a cada momento en frecuencia. FHSS es utilizado para distancias cortas, en aplicaciones por lo general punto a multipunto, donde se tienen una cantidad de receptores diseminados en un área relativamente cercana al punto de acceso.

La señal portadora cambia de frecuencia. o salta (hops), de acuerdo a una secuencia pseudo aleatoria llamada PN o Pseudo Noise. Este es una lista de frecuencias a la cual la portadora brinca en un intervalo de tiempo especificado, antes de repetir el patrón de la secuencia primero.

- Espectro extendido en secuencia directa (DSSS)

DSSS genera un patrón de bits redundante para cada bit que sea transmitido. Este patrón de bit es llamado código chip. Entre más grande sea este chip, es más grande la probabilidad de que los datos originales puedan ser recuperados (pero, por supuesto se requerirá más ancho de banda). Sin embargo, si uno o más bits son dañados durante la transmisión, técnicas estadísticas embebidas dentro del radio transmisor podrán recuperar la señal original sin necesidad de retransmisión. DSSS se utilizará comúnmente en aplicaciones punto a punto.

Para que estas tecnologías se puedan utilizar existen algunos dispositivos fundamentales que se deben tener en consideración, es por ello que en el siguiente tema se describirán en términos generales los más importantes.

## 2.8 Dispositivos de la infraestructura inalámbrica

Existen algunos dispositivos físicos que se tienen que considerar antes de implementar una red inalámbrica. En la arquitectura inalámbrica, existen componentes que forman toda la infraestructura inalámbrica, enseguida se estudiarán los siguientes:

- El medio
- Antenas
- Estaciones Móviles
- Puntos de acceso.

- **El medio.** En todo sistema inalámbrico existe un medio físico o también conocido como medio de distribución. En otras palabras el medio es la infraestructura cableada o con mayor capacidad de transmisión que se identifica como la columna de cualquier red o sistema inalámbrico.
- **Antenas.** Es el componente físico que radia la señal modulada a través del aire, sus características más importantes son: su patrón de propagación, ganancia, y potencia de transmisión.
- **Estaciones Móviles.** Se refiere al dispositivo final, donde el usuario obtiene información de la red. Ejemplos de estos dispositivos son las Computadoras portátiles (Laptops), PDA's, teléfonos celulares etc.

Como último punto referente a las redes inalámbricas se estudiarán en términos de extensión su clasificación, tecnologías, protocolos y estándares usados.

### 2.9 Tecnologías Inalámbricas, Protocolos y Estándares

Básicamente las redes inalámbricas se clasifican en 3 categorías, como lo muestra la figura 21:

- Redes PAN (Personal Area Networks)
- Redes LAN (Local Area Network)
- Redes WAN (Wide Area Network)

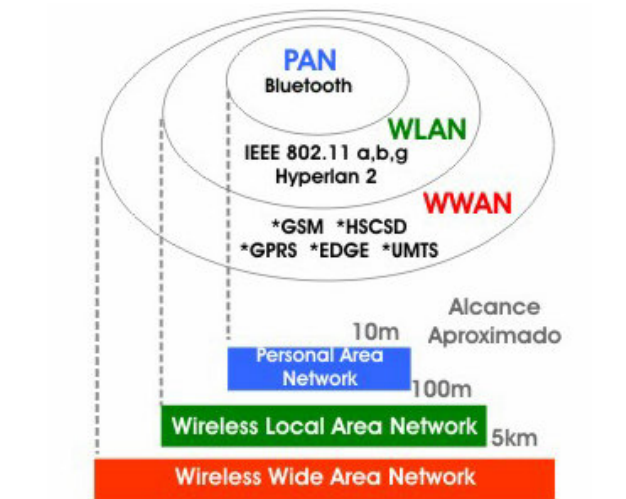


Figura 21.- Clasificación por Extensión de Redes Inalámbricas

Las Redes PAN (Personal Area Networks) cuyo alcance y velocidad de transmisión de datos es muy pequeña pero útil para intercambiar información entre dispositivos portátiles, como Laptops, celulares entre otros. El Estándar usado comúnmente es 802.15.

Los estándares WLAN son desarrollados por organismos reconocidos internacionalmente, tal es el caso de la IEEE (Institute of Electrical and Electronics Engineers) y la ETSI (European Telecommunications Standards Institute). Una vez desarrollados se convierten en la base de los fabricantes para desarrollar sus productos.

Entre los principales estándares se encuentran:

- **IEEE 802.11:** El estándar original de WLANs que soporta velocidades entre 1 y 2 Mbps.

- **IEEE 802.11a:** El estándar de alta velocidad que soporta velocidades de hasta 54 Mbps en la banda de 5 GHz.
- **IEEE 802.11b:** El estándar dominante de WLAN (conocido también como Wi-Fi) que soporta velocidades de hasta 11 Mbps en la banda de 2.4 GHz.

En la tabla 10 se pueden ver algunas características de los estándares inalámbricos WLAN.

Estándar	Velocidad máxima	Interface de aire	Ancho de banda de canal	Frecuencia
<b>802.11b</b>	11 Mbps	DSSS	25 MHz	2.4 GHz
<b>802.11<sup>a</sup></b>	54 Mbps	OFDM	25 MHz	5.0 GHz
<b>802.11g</b>	54 Mbps	OFDM/DSSS	25 MHz	2.4 GHz

**Tabla 10.-** Estándares Inalámbricos WLAN

Las redes WAN (Wide Area Network). Son todas aquellas redes que se pueden comunicar en distintas ciudades e incluso distintos países, abarcan desde 100 km hasta los 100000 km de distancia aproximadamente. El estándar que está en progreso es 802.20.

Desde la aparición de la primera computadora se ha crecido enormemente hasta nuestra época, se han desarrollado diferentes tecnologías para poder comunicarnos de una manera más fácil y rápida, compartiendo información en todo momento y la computadora se volvió una herramienta de trabajo que facilita estas tareas.

Desgraciadamente los avances científicos y tecnológicos se han suscitado por eventos bélicos, y siempre tratando las potencias mundiales de estar al frente en tecnología para cualquier suceso futuro. Si recordamos la primera computadora se construyó para desarrollar misiles (Z3) o el COLUSSOS para descifrar códigos secretos, etc.

Actualmente el uso de la computadora no sólo es de uso militar sino empresarial o simplemente personal con lo que se comparte información muy importante por la red, en la cual pueden estar conectados miles de usuarios, es por ello que la red se vuelve insegura y muy peligrosa. De esta manera se desarrollan algoritmos que puedan proteger esta información, en el siguiente capítulo se estudiará el tema de la seguridad en las redes de datos.

## Capítulo 3

### Seguridad Informática

Al desarrollarse las redes de datos existieron miles de aplicaciones creadas por los programadores, aprovechando el poder compartir recursos, acceso a información a larga distancia sin estar presente físicamente, sin embargo, se presentaron nuevas necesidades, entre las más importantes el tema de seguridad, ya que hay información sumamente valiosa, militar, empresarial o simplemente datos personales de un usuario. De aquí nace esta enorme necesidad y no es nueva, ya que desde la época antigua se buscaban métodos para ocultar la información. En el siguiente capítulo se desmenuzará este tema iniciando con algunos conceptos básicos para estar familiarizado con algunos conceptos, posteriormente se revisará a través de la historia y poder ver cómo se han desarrollando diferentes mecanismos de seguridad hasta llegar hasta lo que hoy se maneja para la protección de la información.

#### Estados de la Información

El término Tecnología de Información se encuentra presente desde hace mucho tiempo y se refiere a que tecnologías se aplican para el uso y manejo de la información. Enseguida se revisarán los diferentes posibles estados en los que puede estar la información.

La información puede estar en cuatro estados: adquisición, creación, almacenamiento y transmisión. Y cada uno de estos estados tiene cuatro propiedades de seguridad: confidencialidad, integridad, autenticidad y disponibilidad.

En el caso de los seres humanos podemos manejar la información mediante los 5 sentidos. La vista, el olfato, el tacto, el gusto y el oído.

Por ejemplo en el medio ambiente y la naturaleza, podemos adquirir información mediante los 5 sentidos y durante este proceso pueden estar los cuatro estados antes mencionados (adquisición, creación, almacenamiento y transmisión).

Una vez adquirida, permanece en memoria donde se puede olvidar (disponibilidad), tergiversar (integridad), divulgar intencionalmente (confidencialidad) y confundir (autenticidad).

Estos estados de la información son la base para su estudio y después de conocerlas y comprenderlas podemos estudiar una a una para buscar mecanismos que puedan protegerla en cada estado. En el siguiente tema se hablará de esto.

#### 3.1 Seguridad de la Información

Entrando propiamente en el tema de seguridad se iniciará mostrando sus características. Las características que definen la seguridad de la información son las siguientes:





**Buenas Prácticas.** Es necesario contar con normas y procedimientos para la correcta vigilancia del manejo de la información. Las buenas prácticas nos ayudan a reaccionar con efectividad ante cualquier emergencia informática.

**Mecanismos.** Existe una gran diversidad de mecanismos, algunos muy específicos dedicados a mejorar la seguridad informática (Confidencialidad, Integridad, Autenticidad y Disponibilidad).

**Proporcionalidad.** Existe una regla que se debe tomar en cuenta antes de implementar algún mecanismo de seguridad llamado proporcionalidad, indica que no vale la pena usar recursos cuyo costo sea mayor al valor de la información que se está protegiendo.

Si se consideran correctamente estos 5 pilares se habrá dado un paso delante de las personas mal intencionados que intentan apoderarse de información útil, es importante hacer notar que aún estos pilares aplican en todo momento y no sólo para la informática como lo verá en el siguiente tema.

## 3.2 Seguridad Informática

La necesidad de proteger la información contra accesos no autorizados por parte de enemigos reales, potenciales, ha sido una constante en la historia de la evolución de la humanidad. Indiscutiblemente los detonantes que han traído avances científicos y tecnológicos son las guerras, y en lo que respecta a la seguridad jugó un papel muy importante.

Durante La Primera Guerra Mundial se usaron técnicas de ocultamiento de la información y durante La Segunda Guerra Mundial se hicieron avances muy importantes en estas técnicas, utilizando algoritmos de cifrado más complejos. La Segunda Guerra Mundial fue un suceso clave respecto a la seguridad informática, ya que no sólo se usaban técnicas de ocultamiento para fines militares, sino que también se usaron para relaciones diplomáticas, secretos comerciales e industriales y sobre todo almacenar la información de manera segura en las enormes computadoras de la época. El problema se acrecentó para fines de la segunda guerra, cuando las computadoras se hicieron de propósito general multiusuario, ya que actualmente ahora la información no sólo era militar sino de diversa índole y con diferentes aplicaciones (bancarias, nomina, etc.).

Pero para entender como influyeron estos eventos en el tiempo y como desarrollaron algoritmos cada vez más complejos, se examinarán en términos de seguridad las generaciones de las computadoras.

### Generación de computadoras en relación con seguridad informática

Primera Generación de Computadoras (1945 – 1956). Los problemas de seguridad informática surgen con el nacimiento de las computadoras. Durante la primera generación de computadoras (1945-1956) el problema de seguridad consistía en el personal que tenía acceso a las máquinas, se requería un nivel de especialización alto, y las computadoras eran de propósito específico difíciles de manipular orientadas al hardware, podían ejecutar una tarea a la vez (monotarea) y un solo proceso a la vez (monoproceso).

Segunda Generación de Computadoras (1956 -1963). Durante la segunda generación de computadoras (1956 -1963) existen avances importantes provocando problemas en la seguridad de la información, las computadoras ya pueden ejecutar más de un proceso a la vez (multi proceso) mediante tiempo compartido, esto hace que al compartir recursos como memoria, procesador algún programa malicioso podría enterarse de los datos y operaciones que se están ejecutando “simultáneamente”.

Otro problema es que el personal que hace uso de las computadoras ya no es tan especializado, ya que ahora la alimentación de instrucciones ya no es orientada a hardware sino al uso de tarjetas perforadas a través de programas que surgieron como FORTRAN y COBOL.

Tercera Generación de Computadoras (1964 – 1971). En el periodo de la tercera generación (1964 – 1971) surge un enorme problema para la seguridad de la información, de hecho podemos llamarlo como la Primera Revolución de Seguridad, la interconexión de terminales tontas vía líneas telefónicas (Teleproceso), al estar conectadas estas terminales ya no solo se debe cuidar la información en forma local sino también cuando viaja, además como ya se comparten recursos y se conectan varios usuarios a la vez (multiusuario) pues es muy vulnerable la información.

Cuarta Generación de Computadoras (1971 – Presente). En la cuarta generación de computadoras (1971 – presente) surge lo que podemos llamar la Segunda Revolución de Seguridad, la aparición de la PC, esto trajo grandes dificultades para la seguridad de la información por lo siguiente:

1. Ninguna PC fue diseñada pensando en el tema de seguridad.
2. No todos los usuarios son expertos en cómputo, por lo que el mal uso de los equipos frecuentemente compromete la seguridad de la información.
3. La amplia disponibilidad de PC'S pone la capacidad de amenaza en las manos de más personas, incrementando el riesgo.

Como Tercera Revolución de Seguridad se tiene el surgimiento de Internet, con lo que crecen exponencialmente los problemas referentes a la seguridad de información, ya que ahora tenemos millones de usuarios conectados en red, compartiendo recursos y en la mayoría de los casos no hay una política de seguridad que los administre.

Algunos problemas de seguridad inherentes a Internet son:

- Facilidad de curiosear
- Servicios de TCP/IP Vulnerables
- Complejidad en la configuración.
- Códigos Ejecutables (código malicioso, virus)
- Navegador inseguro
- Facilidad de acceso a recursos
- Deficiente control de calidad en los productos de software

Tal como se revisó, las necesidades son muchas, ya que existen muchas maneras de hacer daño, enseguida se revisará el tema de amenazas, ataques y vulnerabilidades.

## Vulnerabilidades, amenazas y ataques

Para poder estudiar la seguridad se utilizará un modelo conceptual mostrado en la figura 22 y así entender mucho mejor los términos que se utilizarán para la mejor comprensión de las ideas plasmadas en el presente trabajo.

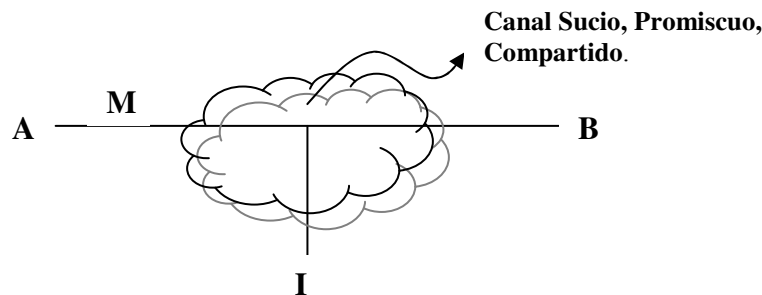


Figura 22.- Modelo conceptual de un canal de comunicación

El objetivo es estudiar como se hace para que **A** envíe un mensaje (**M**) a **B** por un medio de transmisión inseguro donde un intruso (**I**) puede interceptar, modificar, o sólo leer la información.

Definamos algunos términos para estar familiarizados.

- **Sistema de Cómputo.** Es el conjunto formado por la colección de hardware, software, medios de almacenamiento, datos o información y personas involucradas en el conjunto.
- **Compromiso de Seguridad.** Comprometer la seguridad de un sistema equivale a la posibilidad de provocar pérdida o daño al sistema.
- **Vulnerabilidad.** Es cualquier debilidad que pueda explotarse para causar pérdida o daño al sistema.
- **Ataque.** Es cualquier acción que explota la vulnerabilidad del sistema de cómputo. En general se clasifican en Activos y Pasivos.
  - Pasivo. Consiste en sólo observar comportamientos leyendo información sin alterarla, sólo se pierde la confidencialidad y privacidad de los activos del sistema. Generalmente son la antesala para un ataque activo.
  - Activo. Consiste en la capacidad de interceptar los activos del sistema, modificarlos o borrarlos, eso quiere decir que afecta la autenticación e integridad de la información.
- **Amenaza.** Cualquier circunstancia con el potencial suficiente para causar pérdida o daño al sistema.

Existen cuatro tipos de amenazas principales a los sistemas que explotan las vulnerabilidades de los activos en el sistema. Estas amenazas son: interrupción, interceptación, modificación y fabricación. Como lo muestra la figura 23.

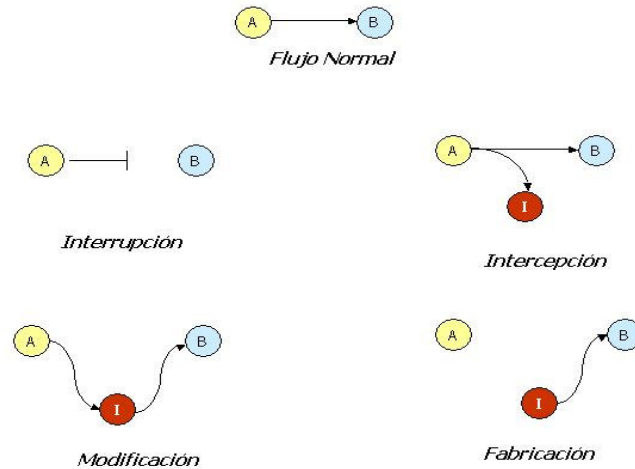


Figura 23.- Tipos de Amenazas

- **Interrupción.** Un activo del sistema se pierde, se hace no disponible o inutilizable. Un ejemplo es un borrado de un programa.
- **Intersección.** El intruso puede ver el activo del sistema sin tener acceso pero no causa daño (al menos en ese momento). Un ejemplo es un Sniffer.
- **Modificación.** El intruso logra el acceso al activo del sistema, lo puede manipular y modificar. Un ejemplo es acceder a una base de datos modificando datos críticos del sistema.
- **Fabricación.** El intruso puede fabricar mensajes haciéndose pasar por algún proceso autorizado. Un ejemplo es agregar registros a una base de datos.

**Servicios y Mecanismos de Seguridad.** Definamos qué es un servicio de seguridad y qué es un mecanismo de seguridad.

- Servicio de Seguridad. Es una característica que debe tener un sistema para satisfacer una política de seguridad.
- Mecanismo de Seguridad. Es un procedimiento concreto utilizado para implementar el servicio de seguridad.

El estándar ISO 7498 -2 identifica cinco clases de servicios de seguridad: confidencialidad, autenticación, integridad, control de acceso y no repudio. En la tabla 12, se colocan los servicios y sus mecanismos de seguridad.

Servicio de Seguridad	Mecanismo de Seguridad
Confidencialidad	Cifrado
Autenticación	Protocolos criptográficos
Integridad	Funciones Hash
Control de Acceso	Modelos matriciales Bell LaPadula
No Repudio	Firma Digital

Tabla 12.- Servicios de Seguridad y su correspondiente Mecanismo de Seguridad

Para entender con mayor facilidad los servicios de seguridad, se utilizará nuevamente el modelo conceptual de la figura 22, el cual por razones didácticas se colocará nuevamente enseguida.

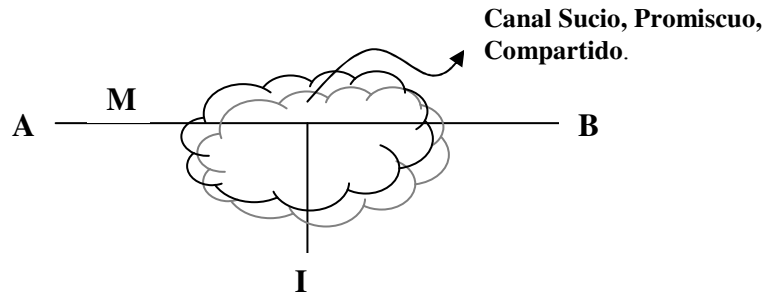


Figura 22.- Modelo conceptual de un canal de comunicación

**Confidencialidad.** Consiste en garantizar que la información sólo pueda ser accedida por las partes autorizadas (Solo A y B).

**Autenticación.** Consiste en garantizar que en una comunicación las entidades participantes sean las quien dicen ser (A y B).

La autenticación se puede clasificar de tres formas como lo muestra el cuadro 5:

<b>Autenticación</b>	<b>Como Proceso de Identificación</b>	<ul style="list-style-type: none"> <li>- <b>En algo que se sabe.</b> (Contraseña, clave)</li> <li>- <b>En algo que se tiene.</b> (Llave física, metal)</li> <li>- <b>En algo que se es.</b> (DNA, iris, huella digital)</li> </ul>
	<b>Quienes participan en la Identificación</b>	<ul style="list-style-type: none"> <li>· <b>Directa.</b> (Solo A y B participan)</li> <li>· <b>Indirecta.</b> (Cuando A y B confían en un tercero)</li> </ul>
	<b>Quien se Autentica</b>	<ul style="list-style-type: none"> <li>· <b>Mutua.</b> (Se autentican ambos A y B,)</li> <li>· <b>Unilateral.</b> (Solo se autentica A y no B)</li> </ul>

Cuadro 5.- Clasificación de la Autenticación

Como proceso de identificación

1. En algo que se sabe. Se basa en palabras que se memorizan como claves o contraseñas.
2. En algo que se tiene. Se refiere a algo que se posee como alguna moneda o una llave física (metal) o algún objeto tangible.
3. En algo que se es. Alguna característica inherente a la entidad que se autentica, como huellas digitales, su voz, etc.

Quienes participan en la identificación

- Directa. Es directa si en el proceso de autenticación sólo intervienen las partes interesadas (Solo A y B participan).
- Indirecta. Es indirecta si en el proceso de autenticación interviene una tercera parte confiable que actúa como autoridad o juez que avala la identidad de las partes (Cuando A y B confían en un tercero).

Quien se autentica

- Mutua. Será mutua cuando se requiere que ambas partes se autenticuen entre sí (Deben autenticarse ambos A y B).
- Unilateral. Será unilateral si sólo una de las partes se autentique con la otra y no es necesario que la otra se autentique ante la primera (Sólo se autentica A y no B, esto puede ser riesgoso).

**Integridad (Verificación).** Consiste en proteger los activos del sistema contra modificaciones, alteraciones, inserciones, borrado o alguna acción que atente la integridad de los activos.

**Control de Acceso.** Consiste en proteger los activos del sistema contra accesos no autorizados.

**No Repudio.** Proporciona la protección contra la posibilidad de que alguna de las partes involucradas en una comunicación niegue haber enviado o recibido algún mensaje, u originado o haber sido destinatario de una acción.

Es importante conocer de qué tratan o que estudian estos mecanismos, sin embargo, existe toda una ciencia que se encarga precisamente del ocultamiento de la información, la criptología, la cual se estudiará en el siguiente punto.

### 3.3 Criptología y Criptografía

La Criptología [19] es la ciencia genérica de ocultamiento de información y se divide en dos áreas: la Criptografía y el Criptoanálisis.

**La Criptografía** busca métodos para asegurar la confidencialidad y/o integridad de los mensajes, transformándolos de tal manera que sólo los poseedores del elemento clave de transformación, la llave, pueden “destransformarla”.

**El Criptoanálisis** en cambio busca romper los métodos que la criptografía empleó sin conocer la llave, o bien construir señales codificables aceptadas como auténticas.

**Criptografía.** La Criptografía es el arte de enmascarar u ocultar los mensajes con métodos diversos, que cobran sentido cuando se conoce la llave secreta. Desde la antigüedad el hombre ha hecho gala de su ingenio para garantizar la confidencialidad de sus comunicaciones. Actualmente ya es considerada como ciencia aplicada, una rama de las matemáticas, debido a su relación con otras ciencias como la teoría de números, la estadística, y las teorías de la información y de información computacional.

Un ejemplo de ello es el método por sustitución que empleó Julio Cesar, donde a cada letra hacía un corrimiento de tres posiciones más allá del alfabeto. Actualmente se conoce como  $C=M+K$ , donde **M** representa el texto en claro, **K** es la llave secreta (Julio Cesar utilizó  $K=3$ ) y **C** representa la correspondiente letra cifrada.

Durante la Segunda Guerra Mundial, la Criptografía jugó un papel sumamente importante empleando métodos por sustitución en máquinas cifradoras / descifradoras, cada uno de los más grandes ejércitos empleaban sus propias versiones: la SIGABA de los norteamericanos, la Púrpura de los japoneses, la TYPEX de los ingleses, y la famosa ENIGMA de los alemanes.

Esta última, fue el talón de Aquiles para los Alemanes ya que fueron descifrados sus mensajes militares por los norteamericanos.

En los años setenta, DES (Data Encryption Standard), un sistema estandarizado a partir del algoritmo Lucifer de IBM hecho por los norteamericanos debido a la creciente demanda de empresas e instituciones de todo tipo, para después ser desechado por considerarlo poco seguro.

Philip Zimmermann, un criptógrafo aficionado, desarrolló un sistema criptográfico aparentemente inviolable en 1917, el P.G.P. y lo distribuyó por las redes de comunicación para que cualquiera pudiera utilizarlo: ahora es uno de los más comunes en cuestión de correo electrónico.

La criptología es una lucha constante de tratar de mantener secreta la información y el enemigo que trata de descifrarla, es por ello que Kerckhoff propone algunas reglas.

### Reglas de Kerckhoff

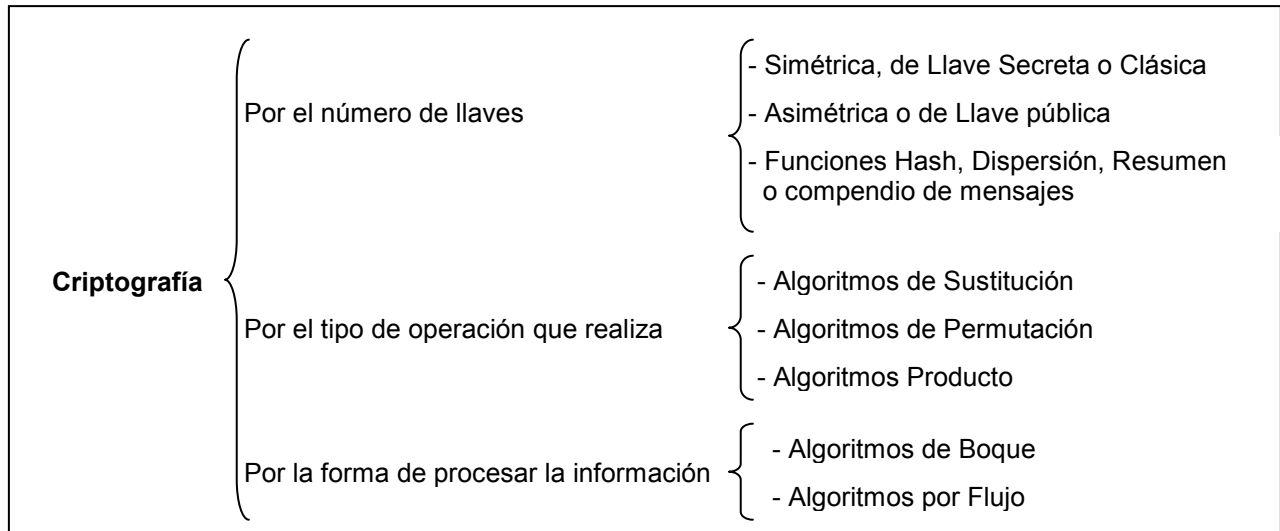
El holandés Kerckhoff hizo algunas recomendaciones que debían cumplir los sistemas criptográficos:

- No debe existir ninguna forma de recuperar mediante el criptograma (**C**) el texto inicial (**M**) o la llave (**K**). Esta regla se considera cumplida siempre que la complejidad del proceso de recuperación del texto original **M** sea suficiente para mantener la seguridad del sistema.
- Un sistema criptográfico debe estar compuesto por dos tipos de información:
  - Pública: como es la familia de algoritmos que lo definen.
  - Privada: la llave utilizada para cada cifrado en particular.
- La forma de escoger la clave debe ser fácil de recordar y modificar.
- Debe ser factible la comunicación del criptograma **C** por los medios de transmisión habituales.
- La complejidad del proceso de recuperación del texto original **M** debe corresponderse con el beneficio obtenido.

Para continuar con el desmenuzamiento de la criptografía se revisará la forma en cómo se clasifica.

### Clasificación de la Criptografía

La Criptografía se clasifica de acuerdo a 3 bases principales y se muestra en el cuadro 6:



**Cuadro 6.-** Clasificación de la criptografía

En seguida se describirá a qué se refiere cada punto de la clasificación:

**Simétrica, de Llave Secreta o Clásica.** Los criptosistemas de llave secreta, privada, o clásicos, se basan en que el emisor y el receptor comparten una única llave secreta  $k$ ; de forma que el proceso de encriptación  $E$  es el inverso del de descryptación  $D$ , y el conocimiento de uno de ellos permite el conocimiento del otro con facilidad.

Por eso también se conocen como sistemas simétricos. A lo largo de la historia, han sido muchos y muy utilizados, pero entrañan un problema: la clave utilizada ha de transmitirse en algún momento entre el emisor y el receptor, por lo que se requiere un canal seguro, es decir, un canal de comunicación donde no pueda existir ningún intruso, lo que en la práctica es imposible. La seguridad del sistema depende del secreto de la llave, y son, generalmente, más fáciles de criptoanalizar que los de llave pública.

Algunos algoritmos que se basan en este tipo de criptografía son los de sustitución y transposición, tales como: el César, Vernam, Vigenere y algunos algoritmos modernos como DES y AES.

**Asimétrica o de Llave Pública.** A diferencia de los algoritmos de llave privada o simétrica, se basan en que cada usuario posee un par de llaves, la primera de las cuales se hace pública, y es la que utiliza otro usuario para transmitir un mensaje, mientras que la segunda permanece privada y sólo es conocida por su dueño, permite recuperar los mensajes cifrados.

La llave pública se usa para cifrar y la llave privada para descifrar, ambas llaves están relacionadas por su origen y su uso, pero del conocimiento de la llave pública no se puede derivar la llave privada.

De esta manera, se puede crear un directorio, con todas las claves públicas de los usuarios que participan en un sistema determinado.

La figura 24 ilustra como trabaja este algoritmo:



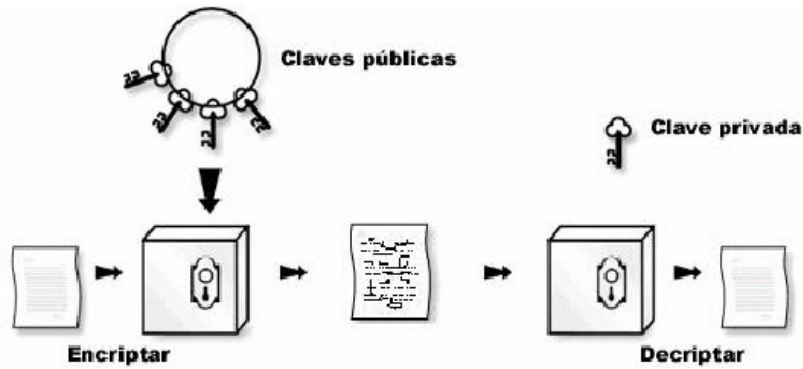


Figura 24.- Asimétrica o de Llave Pública

Dentro de esta clasificación se encuentran algoritmos como RSA en honor a sus autores Rivest, Sharmir y Adleman; es uno de los métodos más utilizados.

**Funciones Hash, Dispersión o compendio de mensajes.** Este tipo de criptografía garantiza la integridad del mensaje realizando una transformación mediante una función, en las siguientes figuras consecutivas, 25, 26, 27 y 28 respectivamente se ejemplifica como trabaja una función Hash:

El código ASCII asigna un número a cada letra o signo de puntuación

65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/

Figura 25.- Ejemplo de Función Hash

Podemos substituir cada letra de un texto por su código ASCII

E	n		u	n		r	i	n	c	ó	n		d	e	
69	110	32	117	110	32	114	105	110	99	243	110	32	100	101	32
l	a		M	a	n	c	h	a		d	e		c	u	y
108	97	32	77	97	110	99	104	97	32	100	101	32	99	117	121
o		n	o	m	b	r	e		n	o		q	u	i	e
111	32	110	111	109	98	114	101	32	110	111	32	113	117	105	101

Figura 26.- Ejemplo de Función Hash

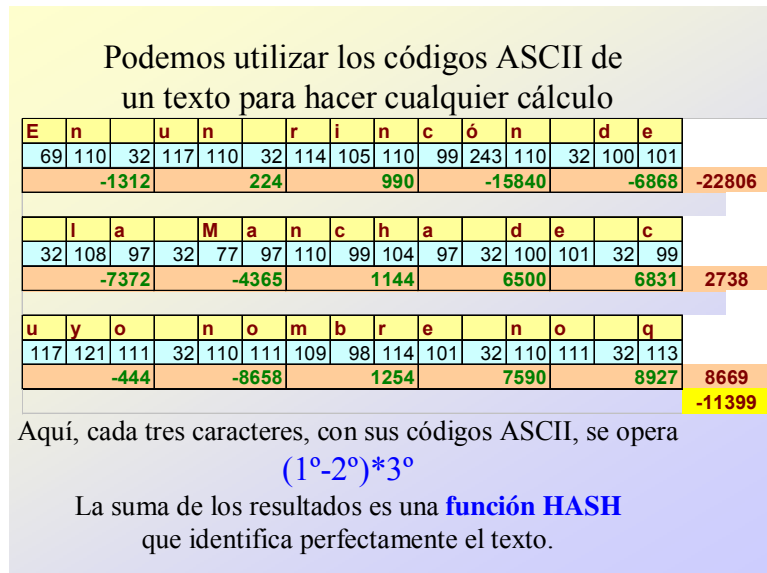


Figura 27.- Ejemplo de Función Hash

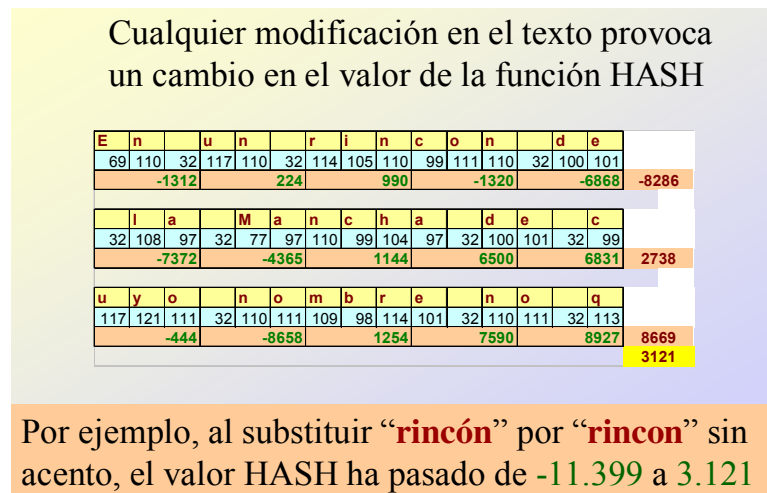


Figura 28.- Ejemplo de Función Hash

**Algoritmos de Sustitución.** Este algoritmo se basa en substituir cada elemento del mensaje por un elemento del texto cifrado vía una regla de una llave. El ejemplo claro es que utilizaba Julio César donde hacía un corrimiento en tres posiciones en el alfabeto, representado actualmente como  $C=M+K$ , donde **M** representa el texto en claro, **K** es la llave secreta y **C** representa la correspondiente letra cifrada. Un ejemplo más es el que utiliza el algoritmo de Vigenere, que utiliza substitución poli alfabética.

**Algoritmos de Permutación.** Este algoritmo también llamado de transposición hace un reordenamiento de las posiciones del texto en claro, resultando un texto cifrado.

**Algoritmos Producto.** Este algoritmo combina tanto el algoritmo por substitución como el de permutación o transposición logrando un mejor cifrado. A esta clasificación pertenecen la mayoría de los algoritmos simétricos modernos como AES entre otros.

**Algoritmos de Bloque.** Estos algoritmos dividen el mensaje en bloques de tamaño fijo, aplicando operaciones de cifrado y descifrado sobre cada bloque, al final el mensaje cifrado es la concatenación de todos los bloques cifrados.

**Algoritmos por Flujo.** Este algoritmo procesa el mensaje como un todo unido. Para cifrar un mensaje completo se debe aplicar la operación de cifrado a todos los componentes como un todo, es importante hacer notar que se requieren más recursos físicos (memoria de la computadora), ya que hay que subir todo el mensaje a memoria y aplicar la operación para cifrar.

La criptografía tal como observamos contribuye con diferentes algoritmos que tienen el mismo objetivo "Ocultar la Información", sin embargo en el mundo de la informática en ocasiones no es necesario ocultarla del todo, sino por ejemplo nos interesa la Integridad de la información, es decir no importa que la puedan visualizar, pero lo que interesa es que no se pueda modificar en el trayecto, en fin pueden surgir diversas necesidades, pero enseguida se verá como resolver algunos mediante la firma y huella digital.

### 3.4 Firma Digital

El proceso de firma digital [20] es equivalente a un cheque bancario el cual tiene las siguientes características:

1. Un cheque es un objeto tangible que autoriza una transacción bancaria.
2. La firma del cheque confirma la autenticidad del firmante legítimo.
3. En caso de falsificación, una tercera parte puede intervenir para juzgar la autenticidad.
4. Un cheque puede ser cancelado de tal modo que no se pueda rehusar.
5. El papel del cheque es inalterable.

La traslación al mundo digital sería la siguiente:

1. **A** envía a su banco un mensaje autorizando una transferencia a **B**.
2. El banco de **A** debe ser capaz de verificar y probar que el mensaje viene de **A**.
3. El banco también debe asegurarse de que el mensaje es íntegramente el que **A** envió.
4. **A** desea asegurarse de que el banco no pueda falsificar su mensaje.
5. Ambas partes, **A** y el banco, desean asegurarse de que el mensaje es nuevo, que no es un rehúso del mensaje previo y que no haya sido alterado.

Una firma digital consiste en una transformación por medio de una función de firma, que relaciona de forma única el documento o archivo con esa función de firma y un elemento propio de la identidad del firmante que es una llave de firma (esta llave debe ser privada).

La firma digital consta de dos partes bien diferenciadas:

1. **Proceso de Firma:** El emisor firma (aplica función de firma) el documento con su llave privada, enviando al destinatario tanto el documento en claro como el firmado (encriptado).

**A** (el firmante) crea una firma digital **s** mediante su llave privada para un mensaje **M**, de la siguiente manera:

- ✓ **A** calcula  $s = S_A(M)$ , donde  $s$  es la firma de **A**. Se obtiene aplicando la función  $S_A$  con su llave privada, sobre el mensaje **M**.
  - ✓ **A** envía a **B** la pareja  $(M,s)$ .
2. **Proceso de Verificación de la Firma:** El receptor aplica la función de verificación al documento firmado con la clave pública de **A** y comprueba que coincide con el documento original, lo que atestigua de forma total que el emisor del mismo ha sido efectivamente **A**.

**B** (el verificador) verifica que la firma  $s$  sobre el mensaje **M** haya sido creada por **A**, mediante su llave pública de la siguiente manera:

- ✓ Obtiene la función de verificación  $V_A$  (llave pública) de **A**.
- ✓ Calcula  $v = V_A(M,s)$
- ✓ Acepta la firma como creada por **A** si  $v = \text{verdadero}$ , y la rechaza si  $v = \text{falso}$ .

El proceso se ilustra en la figura 29:

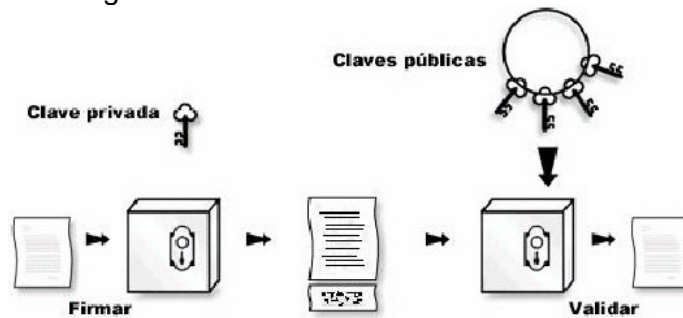


Figura 29.- Proceso de Firma Digital

Si se quisiera firmar digitalmente un documento extenso, se tiene que pensar en que los procesos son lentos y consumen demasiados recursos. Para solventar éste aspecto sólo se firma la huella digital del documento o mensaje, es decir se aplica una función hash al documento a firmar. Esta función hash también debe ser acordada previamente entre el emisor y el receptor.

A un mensaje resumido mediante una función hash y encriptado con una llave privada es lo que en la vida real se denomina firma digital.

El esquema de firma digital mediante una función hash no lo muestra la figura 30:

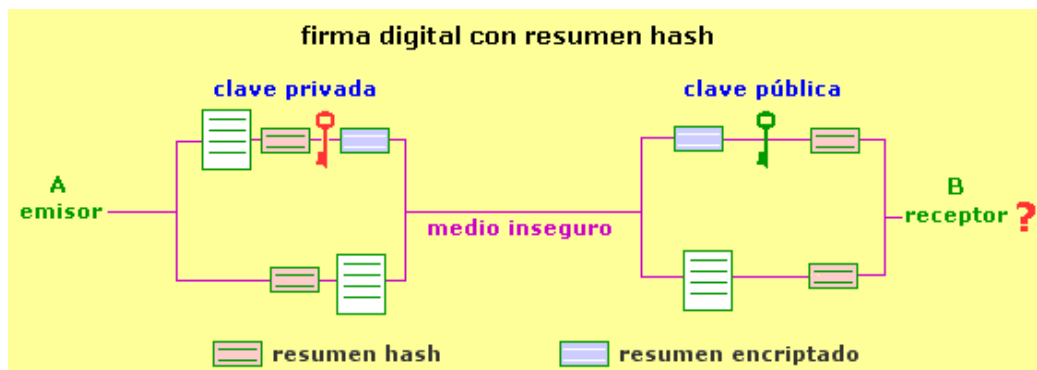


Figura 30.- Firma Digital con resumen Hash

Y su mecanismo es el siguiente:

1. El emisor aplica una función hash conocida al documento, con lo que obtiene un resumen hash del mismo.
2. Firma dicho resumen con su clave privada.
3. Envía al receptor el documento original plano y el resumen hash firmado.
4. El receptor B aplica la función hash al resumen sin encriptar y descripta el resumen encriptado con la llave pública de A.
5. Si ambos coinciden está seguro de que ha sido A el que le ha enviado el documento. Si no coinciden, está seguro de que no ha sido A o de que el envío ha sido interceptado durante el medio de envío y modificado.

El tema de las firmas digitales junto con las funciones hash juegan un papel sumamente importante en el tema de seguridad, sin embargo, existe algo que no se ha considerado, las llaves públicas, ¿Están seguras?, ¿Qué hay con el tema de la suplantación?, ¿En dónde se alojan?. Bueno, pues estas respuestas serán respondidas con el siguiente punto que se estudiará, los certificados digitales.

### 3.5 Certificados Digitales

Siguiendo con la importancia que toma la criptografía de llave pública, surge un nuevo problema; qué pasa si se plantea que ahora **A** desea enviar a **B** un mensaje **M** y desea cifrarlo, para esto **A** debe obtener la llave pública de **B** de un directorio público, **A** toma la llave pública que cree que es de **B**, pero no está seguro que esta llave en realidad pertenezca a **B**.

Una solución es la utilización de un certificado digital [20], el cual debe certificar que la llave pública es de quien dice ser, el certificado debe ser intransferible y no modificable, emitido por una tercera parte de confianza (Autoridad Certificadora), que asocia a una persona o entidad una llave pública. Su autenticidad es garantizada por el hecho de que sólo un organismo confiable pueda crearlo.

Un certificado digital típicamente contiene la siguiente información:

- Identificación del titular del certificado: Nombre, dirección, etc.
- Llave pública del titular del certificado.
- Fecha de validez.
- Número de serie.
- Nombre del emisor (autoridad certificadora que generó el certificado).

Para el correcto manejo de los certificados digitales es necesario considerar los siguientes elementos:

Para emitir un certificado, la AC (Autoridad Certificadora) debe validar previamente la identidad del suscriptor, una vez que ha validado la solicitud, genera el certificado correspondiente y lo firma con su llave privada. La característica principal que tiene un certificado digital es implementar servicios de identificación y autenticación, así como distribuir de manera segura llaves públicas. Sin embargo la seguridad radicará en qué tan confiable sea la AC y los mecanismos que utilice ésta para proteger su llave privada.

Una Autoridad Certificadora es una entidad de confianza, responsable de emitir y revocar los certificados digitales, utilizados en la firma electrónica, para lo cual se emplea la criptografía de llave pública.

Para obtener un certificado digital, debe llenarse una solicitud y la información que se envíe será la llave pública de suscriptor e información adicional que dependerá del tipo de certificado que se solicite. Es decir, el suscriptor debe generar su propio par de llaves (Pública y Privada), guardar de manera segura su llave privada y mandar la pública.

El suscriptor puede almacenar de manera segura su llave privada y utilizar diferentes métodos, ya que de eso depende la seguridad de su información, por ejemplo puede almacenarla en algún dispositivo externo como una tarjeta inteligente, o cifrarlo colocando una contraseña.

El proceso de comprobar la información solicitada del suscriptor cubre vital importancia, ya que un impostor podría estar solicitando un certificado que ligue su llave pública con la identidad de otra persona.

Es tan importante esta verificación que en ocasiones la autoridad certificadora llega a solicitar la presencia física de la persona.

Un ejemplo es VeriSign, la cual es una Autoridad Certificadora con mucha aceptación, y clasifica el nivel de seguridad en clases: la clase 1 y 2, está restringido a personas por lo que no comprueba la pertenencia a alguna empresa u organización.

Un certificado de clase 1 solamente proporciona un nombre y dirección de correo electrónico y no se lleva ninguna verificación de datos de la persona como podría ser su domicilio.

Un certificado de clase 2 sí verifica datos como el nombre dirección e información en bases de datos de terceros, un ejemplo de este tipo de certificados son los que se utilizan para alguna compra por Internet.

Los certificados clase 3 y 4 sólo se otorgan a organizaciones y la autoridad certificadora valida con mayor cuidado datos de la organización, así como algunos estándares de estabilidad financiera.

Desde luego, estos elementos son considerados importantes en el tema de la seguridad de la información. Bueno tal como se ha revisado el tema no es sencillo, pero hay que tomar cartas en el problema, ya que hay información muy valiosa y actualmente el tema es de inversiones, situaciones de estrategias militares, etc.

En el siguiente capítulo se revisará la manera en que actualmente se administra la información en todo el campo de la informática, por supuesto considerando los capítulos anteriores, teniendo especial cuidado en el tema de seguridad para el correcto resguardo de la misma.

Con el uso de las computadoras para el manejo de información existen millones de datos que deben ser correctamente manejados, el siguiente capítulo habla precisamente de la administración de las bases de datos en grandes proporciones.

## Capítulo 4 Bases de Datos y Datawarehouse

El presente módulo tiene el objetivo de estudiar lo que es una base de datos, así como los programas que pueden utilizarse para manejarla (SMBD), enfocándose inicialmente al paradigma relacional y al final entrando un poco a lo que se refiere bases de datos orientadas a objetos.

La explotación óptima de las Bases de Datos depende de dos elementos básicos: la herramienta de explotación y el diseño de la base de datos. En seguida se definirán algunos conceptos básicos.

### 4.1 Conceptos y Definición de sistema manejador de bases de datos (SMBD)

Para iniciar al presente módulo se definirán algunos conceptos importantes que se utilizarán a lo largo del texto.

**Base de datos.** Una base de datos [21] es una colección de datos agrupados con cierto orden, lógica y coherencia por un periodo relativamente largo.

Una base de datos se define como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por un Sistema Manejador de Base de Datos (SMBD).

En seguida se verá el paradigma relacional por lo que se establece la siguiente equivalencia en la Tabla 13:

Definición Genérica	Paradigma Relacional
Tabla	Entidad o Relación
Campo o Columna	Atributo
Registro o Renglón	Tupla

**Tabla 13.-** Equivalencia en el Paradigma Relacional

**Dato.** El dato es la unidad mínima de información que expresa una característica asociada a una entidad u objeto.

Cualquier pieza de información puede ser llamada dato. Este dato debe ser significativo. Los datos son representados por medio de valores como números, cadenas de caracteres, o símbolos que sustenten algún significado.

**Tabla.** La tabla es la estructura que contiene uno o más datos y está compuesta por columnas y renglones.

Es la estructura fundamental de un sistema de administración de bases de datos relacional. Una tabla es un objeto que almacena datos en filas y en columnas. Las filas se denominan registros o tuplas y las columnas campos o atributos.

**Columna.** Las columnas están formadas por dos elementos, su nombre y los valores que se registran en ella. Los nombres de las columnas deben ser representativos de los datos que representan.

Una columna o campo es el componente de una tabla que contiene un elemento específico de información. Por ejemplo, nombres, apellidos, direcciones, ciudades, códigos de productos, valores de productos, etc. En una tabla los campos corresponden a las columnas.

**Renglón.** Un renglón se forma por los valores registrados en cada una de las columnas contenidas en una tabla.

Un renglón o también llamado registro es un conjunto de datos (atributos) acerca de un evento, persona, lugar, o algún otro elemento en una tabla. Por ejemplo, datos personales de un empleado, datos particulares de un producto.

**Un DBMS** (Data Base Management System), Es una poderosa herramienta que permite crear y administrar grandes cantidades de información, de manera segura, eficiente y dotándole de persistencia por periodos largos.

El objetivo primordial de un sistema manejador de base de datos es proporcionar un entorno conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfase entre los usuarios y la base de datos.

Un SMD debe cumplir con tres características importantes:

1. **Almacenamiento Persistente.** Un SMD debe soportar, almacenar grandes cantidades de información y ésta debe existir independientemente del uso que haga de ellos cualquier proceso que los acceda, facilitando en forma eficiente el acceso a las estructuras de datos.
2. **Interfaz de Programación.** Un SMD a través de un programa (software) debe permitir el acceso y modificación de los datos con eficiencia.
3. **Administrador de Transacciones.** Un SMD soporta el acceso concurrente a los datos, es decir, acceso simultáneo de distintas transacciones al mismo tiempo.

En la década de los 80 comenzaron a aparecer numerosos SGBD que se anunciaban como "relacionales". Sin embargo, estos sistemas carecían de muchas características que se consideran importantes en un sistema relacional, perdiendo muchas ventajas del modelo relacional. Como ejemplo extremo de esto, los "sistemas relacionales" eran simplemente sistemas que utilizaban tablas para almacenar la información, no disponiendo de elementos como claves primarias, etc. En 1984, Edgar Frank Codd publicó 12 reglas que un verdadero sistema relacional debería de cumplir. En la práctica algunas de ellas son difíciles de realizar.

## REGLAS DE CODD

**Regla 0.** Para que un sistema se denomine sistema de gestión de bases de datos relacionales, este sistema debe usar (exclusivamente) sus capacidades relacionales para gestionar la base de datos.



**Regla 1: Regla de la Información.** Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico, exactamente de una manera: con valores en tablas. Por tanto los metadatos (diccionario, catálogo) se representan exactamente igual que los datos de usuario.

**Regla 2: Regla del acceso garantizado.** Para todos y cada uno de los datos (valores atómicos) de una BDR se garantiza que son accesibles a nivel lógico, utilizando una combinación de nombre de tabla, nombre de columna y valor de clave primaria.

Cualquier dato almacenado en una BDR tiene que poder ser diseccionado unívocamente. Para ello hay que indicar en qué tabla está, cuál es la columna y la fila (mediante la clave primaria).

**Regla 3: Tratamiento sistemático de valores nulos.** Los valores nulos (que son distintos de la cadena vacía, blancos, 0, etcétera) se soportan en los SMBD totalmente relacionales para representar información desconocida o no aplicable de manera sistemática, independientemente del tipo de datos.

**Regla 4: Catálogo dinámico en línea basado en el modelo relacional.** La descripción de la base de datos se representa a nivel lógico de la misma manera que los datos normales, de modo que los usuarios autorizados pueden aplicar el mismo lenguaje relacional a su consulta, igual que lo aplican a los datos normales.

**Regla 5: Regla del sublenguaje de datos completo.** Un sistema relacional debe soportar varios lenguajes y varios modos de uso de terminal (Ej.: rellenar formularios, etc.). Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante una sintaxis bien definida, como cadenas de caracteres y que sea completo, soportando:

- Definición de datos
- Definición de vistas
- Manipulación de datos (interactiva y por programa)
- Limitantes de integridad
- Limitantes de transacción (iniciar, realizar, deshacer) (Begin, commit, rollback).

Además de poder tener interfaces más amigables para hacer consultas, etc. Siempre debe de haber una manera de hacerlo todo de manera textual, que es tanto como decir que pueda ser incorporada en un programa tradicional. Un lenguaje que cumple esto en gran medida es SQL.

**Regla 6: Regla de actualización de vistas.** Todas las vistas que son teóricamente actualizables se pueden actualizar por el sistema.

**Regla 7: Inserción, actualización y borrado de alto nivel.** La capacidad de manejar una relación base o derivada como un solo operando se aplica, no sólo a la recuperación de los datos (consultas), sino también a la inserción, actualización y borrado de datos.

El lenguaje de manejo de datos también debe ser de alto nivel (de conjuntos).

Algunas bases de datos inicialmente sólo podían modificar las tuplas de la base de datos de una en una (un registro de cada vez).

**Regla 8: Independencia física de datos.** Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuando quiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

**Regla 9: Independencia lógica de datos.** Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuando quiera que se realicen cambios a las tablas base que preserven la información.

**Regla 10: Independencia de integridad.** Los limitantes de integridad específicos para una determinada base de datos relacional deben poder ser definidos en el sublenguaje de datos relacional, y almacenables en el catálogo, no en los programas de aplicación.

El objetivo de las bases de datos no es sólo almacenar los datos, sino también sus relaciones y evitar que éstas (limitantes) se codifiquen en los programas. Por tanto, en una BDR se deben poder definir limitantes de integridad.

**Regla 11: Independencia de distribución.** Una SMDB tiene independencia de distribución. Las mismas órdenes y programas se ejecutan igual en una BD centralizada que en una distribuida.

**Regla 12: Regla de la no subversión.** Si un sistema relacional tiene un lenguaje de bajo nivel (un registro de cada vez), ese bajo nivel no puede ser usado para saltarse (subvertir) las reglas de integridad y los limitantes expresados en los lenguajes relacionales de más alto nivel [una relación (conjunto de registros) de cada vez].

Algunos problemas no se pueden solucionar directamente con el lenguaje de alto nivel. Normalmente se usa SQL inmerso en un lenguaje anfitrión para solucionar estos problemas.

## 4.2 El Enfoque Relacional

El enfoque relacional establece equivalencias entre el enfoque tradicional de bases de datos y la propuesta, como lo muestra la siguiente tabla 13, ya antes revisada:

Se define una tabla o entidad como una relación, una columna como un atributo y un registro como una tupla de la relación. Para diseñar una base de datos con un enfoque relacional existen una serie de protocolos que deben cumplirse, uno de ellos es el modelo Entidad – Relación.

### Modelo Entidad – Relación

El modelo Entidad – Relación es esencialmente gráfico en el que se utilizan cuadros para definir entidades y flechas de diferentes características para definir las relaciones que hay entre ellas. En seguida se describirán los elementos de los que consta este modelo.

Existen tres tipos de elementos en un diagrama Entidad – Relación: Entidades, Atributos y Relaciones.

- **Entidades.** Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc.

Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil.

- **Atributos.** Las características de las entidades en base de datos se llaman atributos

Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante óvalos asociados a las entidades por líneas sólidas.

- **Relaciones.** Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Pero para entender mejor esto, se verá un ejemplo:

Se considerará una empresa que requiere controlar a los vendedores y las ventas que ellos realizan; de este problema determinamos que los objetos o entidades principales a estudiar son el empleado (vendedor) y el artículo (que es el producto en venta), y las características que los identifican son:

<b>Empleado:</b>	<b>Artículo:</b>
Nombre	Descripción
Puesto	Costo
Salario	Clave
R.F.C.	

La relación entre ambas entidades la podemos establecer como Venta.

Bueno, ahora nos falta describir como se representa un modelo E-R gráficamente, la representación es muy sencilla, tal como lo muestra la figura 39:

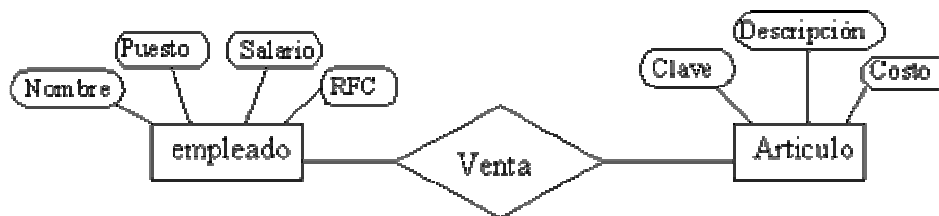


Figura 31.- Ejemplo del Modelo Entidad - Relación

Una característica que tienen las relaciones es su cardinalidad, la cual depende del número de elementos que se relacionen de una entidad a otra, existen tres tipos diferentes de cardinalidad:

- **Una a una (1:1):** Una instancia de **A** está asociada con 1y sólo una instancia de **B**, en ambos sentidos.

- **Una a muchas (1:N):** Una instancia de **A** puede estar asociada con muchas instancias de **B**, pero una instancia de **B** sólo puede estar asociada con 1 instancia de **A**.
- **Muchas a Muchas (N:M):** una instancia de **A** puede estar asociada con muchas instancias de **B** y viceversa.

**Principios de diseño.** Con el fin de dotar de calidad a los diseños de bases de datos, se mencionarán algunos principios indispensables que deben considerarse:

- **Fidelidad.** Sólo se deben incluir elementos que se encuentren dentro de la definición de la aplicación.
- **Eliminación de la redundancia.** No se debe duplicar información en las entidades.
- **Simplicidad.** Para fines prácticos es necesario que el modelado se haga lo más simple posible.
- **Selección de las relaciones adecuadas.** Se debe tener cuidado con la relación que se colocan entre las entidades, ya que de ello depende la eficiencia del diseño.
- **Seleccionar el tipo de elemento adecuado.** Es muy importante seleccionar las entidades adecuadas, las cuales nos representen, lo más cercano posible, los elementos de la realidad.

**Modelado de reglas.** Siguiendo con el diseño de las bases de datos en necesario considerar algunas restricciones o reglas. En el modelo E/R se clasifican las siguientes:

- **Llave Primaria.** Está formada por el atributo o atributos que identifican de manera única cada tupla de la entidad.
- **Valor Único.** Además de la llave primaria, existen atributos que por cuestiones de negocio o de operación no deben repetirse en algunos de sus atributos.
- **Integridad Referencial.** Es el requerimiento de que el valor de algún atributo de la entidad A corresponda al valor existente en la entidad B. El atributo de la entidad A se denomina llave foránea y en la entidad B debe ser una llave primaria.
- **Dominios.** Se refiere a que los valores de los atributos se restrinjan a un conjunto de valores específico.
- **Generales.** Son reglas arbitrarias que deben cumplir los valores de los atributos en una entidad y están dictados por el negocio, o por políticas propias de la aplicación.

Una vez entendido el modelo Entidad – Relación, se pasará al modelo lógico, en el cual se tiene que hacer una traducción y tener algo mucho más cercano a una implementación en algún SMDB.

### Modelo Lógico

Para la traducción del modelo E/R, en el modelo lógico se utilizan símbolos análogos y con este fin se siguen algunas reglas:

1. Contar con un estándar de nomenclatura.
2. El nombre de la entidad debe estar definido en singular y se encuentra fuera de la caja de la entidad.
3. El nombre de los atributos no debe ser largo ni incluir caracteres especiales.
4. Se debe especifica, tipos de datos en forma genérica, aunque éste no se despliegue en el modelo.

5. Cuando el atributo es una llave primaria o forma parte de la llave primaria, se coloca en la parte superior de la caja, separándose del resto de los atributos por una línea.

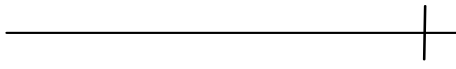
Las entidades se siguen representando en cajas, como lo muestra la tabla 14, con las características antes mencionadas:

**Nombre de la Entidad**

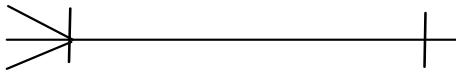
Atributo 1	Tipo de dato
Atributo 2	Tipo de dato
...	...
Atributo n	Tipo de dato n

**Tabla 14.-** De esta manera se representa una entidad.

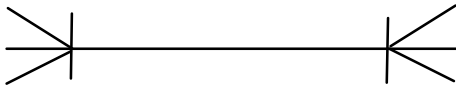
La representación de las relaciones en base a su cardinalidad cambia también de la siguiente manera.



**Relación Uno a Uno.** La línea vertical indica la entidad que hereda su llave

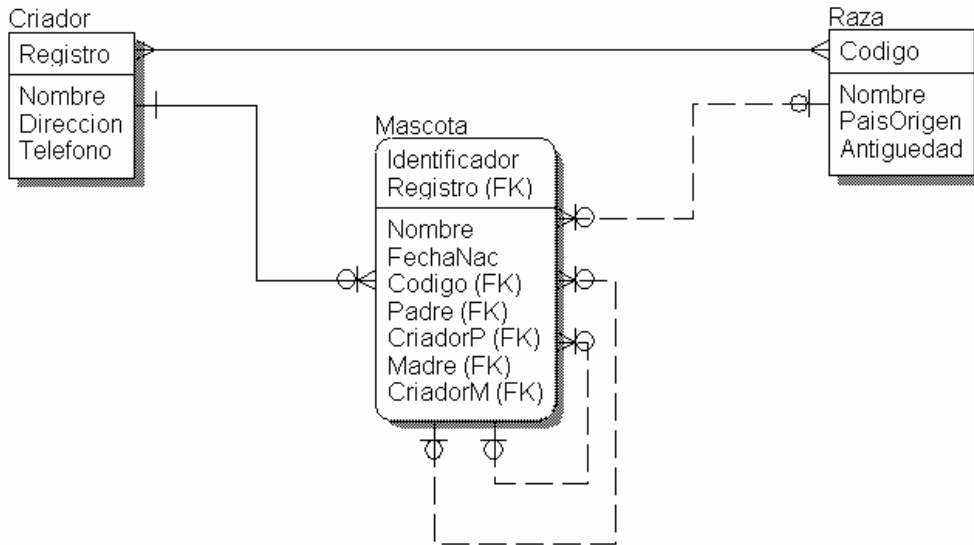


**Relación Uno a Muchos.** La punta triple debe ponerse de lado.



**Relación Muchos a Muchos.** Existe una punta triple en ambos lados.

La figura 32 muestra un ejemplo de un modelo lógico:



**Figura 32.-** Modelo Lógico

**Modelo Físico**

El modelo físico es la representación de la base de datos en un SMDB en específico. Cabe mencionar que no es necesario tener el antecedente del modelo Entidad – Relación y posteriormente el Lógico, ya que se puede diseñar directamente. Por ejemplo, un modelo Físico en Sybase sería como lo muestra la figura 33.

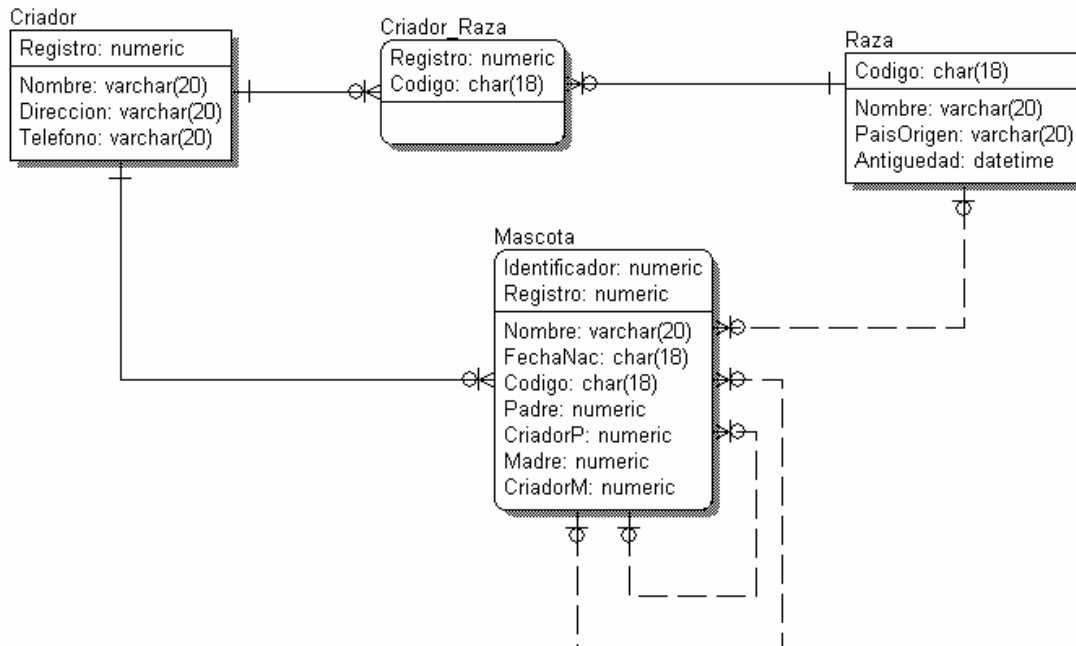


Figura 33.- Modelo Físico en Sybase

La mayoría de las bases de datos actuales todavía están basadas en el sistema tradicional (relacional), sin embargo, la programación orientada a objetos ha crecido enormemente y trae mejoras significativas, es por ello que, poco a poco, las empresas deciden migrar sus aplicaciones. En el siguiente tema se abordarán las bases de datos orientadas a objetos.

### 4.3 Bases de Datos Orientadas a Objetos (BDOO)

La mayoría de las aplicaciones actuales de bases de datos están basadas en tradicionales sistemas manejadores de bases de datos (SMBD).

El modelo de las BDOO [22] es bastante reciente, y propio de los *modelos informáticos orientados a objetos*, trata de almacenar en la base de datos los *objetos* completos.

Para entender aún más el paradigma orientado a objetos se estudiarán los conceptos más importantes:

**Clase.** Es un patrón o plantilla en la que se basan objetos que son similares. Cuando un programa crea un objeto de una clase, proporciona datos para sus variables y el objeto puede entonces utilizar los métodos que se han escrito para la clase. Todos los objetos creados a partir de la misma clase comparten los mismos procedimientos para sus métodos,

En los programas orientados a objetos hay *tres tipos de clases*: clases de control, clases entidad y clases interfaz.

- **Las clases de control** gestionan el flujo de operación de un programa (por ejemplo, el programa que se ejecuta es un objeto de esta clase).
- **Las clases entidad** son las que se utilizan para crear objetos que manejan datos (por ejemplo, clases para personas, objetos tangibles o eventos).
- **Las clases interfaz** son las que manejan la entrada y la salida de información (por ejemplo, las ventanas gráficas y los menús utilizados por un programa).

**Método.** Son operaciones definidas en las clases y que los objetos pueden utilizar, también se utilizan como interfaz de comunicación con otros objetos.

Hay varios tipos de métodos que son comunes a la mayoría de las clases:

- **Constructores.** Un constructor es un método que tiene el mismo nombre que la clase. Se ejecuta cuando se crea un objeto de una clase. Por lo tanto, un constructor contiene instrucciones para inicializar las variables de un objeto.
- **Destruidores.** Un destructor es un método que se utiliza para destruir un objeto. No todos los lenguajes orientados a objetos poseen destructores.
- **Interfaz.** Es un método que devuelve el valor de un atributo privado de otro objeto. Así es como los objetos externos pueden acceder a los datos encapsulados.
- **Mutadores.** Un mutador es un método que almacena un nuevo valor en un atributo. De este modo es como objetos externos que pueden modificar los datos encapsulados.

**Objeto.** Es un elemento autocontenido utilizado por el programa. Los valores que almacena un objeto se denominan atributos, variables o propiedades. Los objetos pueden realizar acciones, que se denominan métodos, servicios, funciones, procedimientos u operaciones.

Una base de datos orientada a objetos incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En el paradigma OO un objeto incluye los datos en las propiedades y los procedimientos en sus métodos. Esta combinación se considera como un paso adelante de la gestión de datos.

El modelo OO soporta relaciones de muchos a muchos, mientras que el modelo relacional requiere resolver dichas relaciones a través de la creación de una nueva entidad.

En este modelo el acceso a los datos es a través de las relaciones que se almacenan en los mismos datos.

Actualmente no hay una tendencia clara de sustitución de los SMBD relacionales a SMBDOO, aunque existen SMBD completamente OO, la mayoría adopta el paradigma OO a través de un híbrido en el sistema relacional.

Por otro lado tal como se indicó en el capítulo 3 (Seguridad Informática), es indispensable que los SMBD cuenten con mecanismos que garanticen la integridad y disponibilidad de la información, es por ello que en el siguiente tema tratará sobre ello.

## 4.4 Seguridad, Integridad y Confidencialidad en las Bases de Datos

La seguridad en las bases de datos tiene que ver con garantizar la disponibilidad, integridad y confidencialidad de los datos.

Se puede agregar seguridad a la base de datos a través de varios mecanismos o niveles para acceder a la información.

**El primer nivel.** Tiene que ver con la redundancia y la disponibilidad, la primera consiste en tener la infraestructura duplicada que garantice la continuidad de la operación, la segunda tiene que ver con respaldos que entren en automático garantizando la integridad de los datos.

**El segundo nivel.** Lo proporciona el SMBD, el cual proporciona niveles de acceso a diferentes usuarios:

- DBA, Administrador de Base de Datos, posee todos los permisos sobre todos los objetos y todas las bases de datos que maneje el SMBD.
- DBO, Es el propietario de la base de datos y posee todos los permisos sobre los objetos de una base de datos en particular.
- User, Es el usuario común que generalmente tiene a un grupo público, sólo tiene permisos de consulta.

**El tercer nivel.** Lo pueden agregar los *TRIGGERS* [Son bloques SQL almacenados asociados a una tabla que se ejecutan o disparan automáticamente cuando se producen ciertos eventos sobre la tabla (inserción, borrado o modificación de fila)] en ellos se puede programar la verificación de los privilegios de modificación de información.

**Un último nivel de seguridad.** Se puede implementar sacrificando el desempeño de la base de datos, pero esto dependerá de qué tan crítica sea la información, implementando protocolos de seguridad o verificación de privilegios.

Desde luego, el tema de las Bases de Datos es muy interesante, ya que muestra el manejo de información, y como a partir de esta se pueden tomar decisiones para las empresas, sin embargo ¿que pasa cuando existen millones de datos dispersos por diferentes bases de datos y una empresa necesita de alguna manera explotarlos para tomar decisiones que reeditarán en dinero principalmente, en estar un paso delante de los competidores?. Bueno pues en el siguiente tema se describirá cómo administrar esos millones de datos mediante un Datawarehouse.

## 4.5 Introducción a Datawarehouse

Hoy día es de vital importancia para las empresas la correcta toma de decisiones, ya que de éstas dependen los resultados de sus negocios e inversiones, para eliminar riesgos e incertidumbre es necesario contar con la información adecuada en el momento adecuado.

La mayoría de las organizaciones hacen lo posible para conseguir buena información, para cumplir con este objetivo dependen fundamentalmente de su arquitectura actual, tanto de software como de hardware. Una de las estrategias de arquitectura tecnológica que brinda beneficios tangibles a las empresas es la Arquitectura Orientada a Servicios (SOA).

Este modulo tiene la intención de abordar las estrategias de negocio más utilizadas y que han reportado mejores resultados para las empresas.



## Evolución del manejo de información

El propósito del desarrollo de los sistemas de información ha sido la recopilación de la información sobre las diferentes áreas de conocimiento del mundo, para auxiliar en la toma de decisiones. Cronológicamente se podría distinguir las siguientes etapas de la evolución de los sistemas de información automatizados:

- Informes por Lotes. La información era difícil de encontrar y analizar. Los repositorios eran poco flexibles y era necesario reprogramar cada cambio en las peticiones.
- Surgen los primeros DSS (Decisión Support Systems) y EIS (Executive Information Systems). Se basaban en terminal, no tenían integración con el resto de las herramientas y aplicaciones.
- Acceso a datos y herramientas de análisis integradas (Intelligent Business Tools). Herramientas de consultas e informes, hojas de cálculo, interfaces gráficas e integradas.
- Surgimiento de las arquitecturas Datawarehouse y Datamart. Se basan principalmente en la organización por capas y en mayor persistencia histórica de la información.

**Una Datawarehouse** es una colección de datos en la cual se encuentra integrada la información de la empresa, vista como un todo y que se usa como soporte para el proceso de toma de decisiones gerenciales.

**Un Datamart** tiene datos específicos de un área o departamento de la organización, esto es, un subconjunto de los datos empresariales, con un valor para el área a la que está orientado.

- Explotación de información, a través de herramientas OLAP (On-Line Analytical Processing). Funcionan sobre la base de datos de Datawarehouse o transaccional, permiten realizar agregaciones y combinaciones mucho más complejas y ambiciosas, con objetivos a nivel estratégico.
- Aparición de las herramientas y técnicas de Minería de Datos y Simulación de escenarios. Permiten la extracción de información con el objetivo de descubrir hechos contenidos en las bases de datos, además tienen muy variadas herramientas que permiten extraer patrones, modelos, tendencias que producen conocimiento.
- Tableros de Control (Balanced Scorecards). Son vistas rápidas de carácter ejecutivo que permiten conocer el estado de la organización mediante el reporte de métricas clave que se definen sobre aspectos críticos para el negocio. Normalmente se visualiza desde cuatro perspectivas: Financiera, Cliente, interna y de Aprendizaje.

En la figura 34 se muestra cómo se relacionan algunos elementos, dando como resultado una estrategia inteligente de negocios.

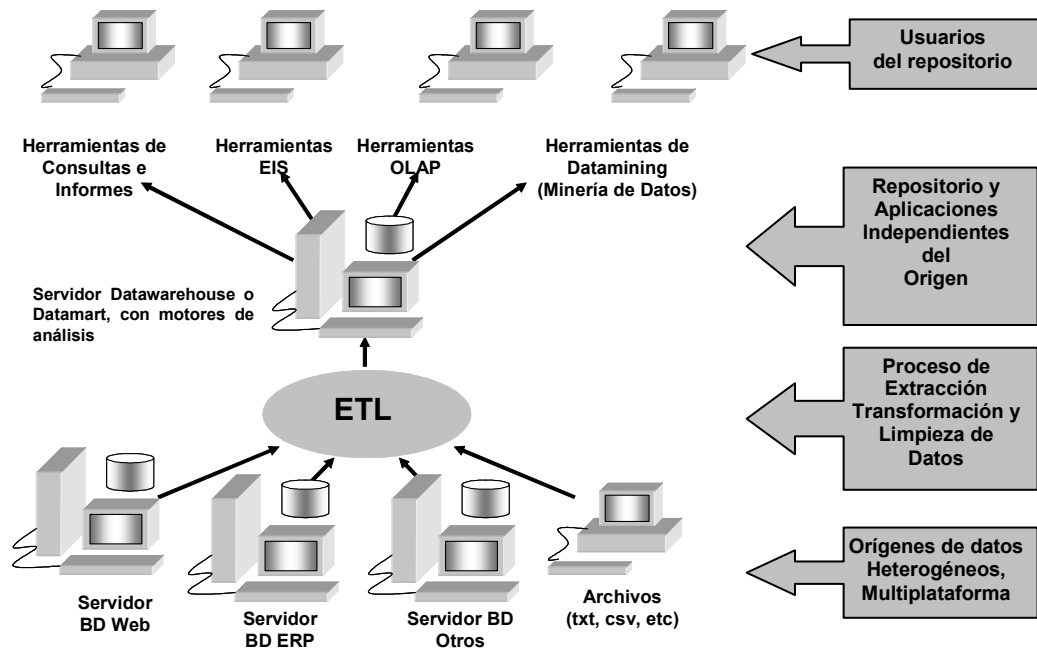


Figura 34.- Arquitectura de Inteligencia de Negocios

Una de las estrategias de arquitectura tecnológica que brinda beneficios tangibles a las empresas es la Arquitectura Orientada a Servicios (SOA), la cual se revisará en el siguiente punto.

### Arquitectura Orientada a Servicios (SOA)

La complejidad que han adquirido las TIC's (Tecnologías de Información y Comunicaciones) es tal, que una empresa requiere de diferentes especialistas responsables para la administración y operación de sus tecnologías. Esto se traduce en altos costos para las empresas.

Por tal motivo han optado cada vez con mayor fuerza por la homogenización de sus TIC's, logrando la reducción de costos y disminuyendo la cantidad de personal.

La estrategia de homogenización no quiere decir que las empresas se casen con una marca en especial, sino que adopten estándares abiertos que permitan la integración de tecnologías de marcas diversas y que garanticen la interoperabilidad.

Por otro lado, es necesario instrumentar una estrategia de organización de la tecnología mediante una arquitectura que permita la delimitación de capas, conformadas por componentes autónomos, pero integrados por un núcleo común que orqueste la operación.

La arquitectura de referencia que cumple con estas características se denomina **SOA** (Service Oriented Architecture) y representa un esfuerzo de las principales organizaciones de comercio electrónico, responsables de la administración de estándares y arquitecturas de servicio Web: OASIS, WC y WS-I.

SOA es el uso de servicios para dar soporte a los requerimientos de negocio e implica la disponibilidad de los servicios, de manera independiente y estandarizada, para cualquier participante de la red empresarial

Conceptualmente la SOA integra componentes desde cinco marcos de referencia básicos:

- **Infraestructura.** Define las directrices para la instalación y configuración de las TIC's bajo especificaciones de conectividad, integridad, escalabilidad, redundancia y alta disponibilidad.
- **Administración.** Abarca la infraestructura propia y la operada por terceros (Outsourcing) con una visión orientada a servicios y basada en las mejores prácticas.
- **Aprovisionamiento.** Toma las redes semánticas como estándar para representar y modelar los componentes existentes en la base de datos única, así como las relaciones que guardan entre ellos.
- **Negocios Electrónicos (e-Business).** Utiliza estándares de organismos como el W3C, ISO y OASIS, entre otros, para temas de colaboración, comercio electrónico, aplicaciones y comunidades virtuales.
- **Núcleo de Integración,** que hace posible la interoperabilidad de los componentes de los cuatro marcos de referencia anteriores, a través de la utilización de estándares para servicios Web, redes semánticas, administradores de procesos de negocio e inteligencia de negocios.

La figura 35 nos muestra un modelo conceptual de la organización de la arquitectura SOA:

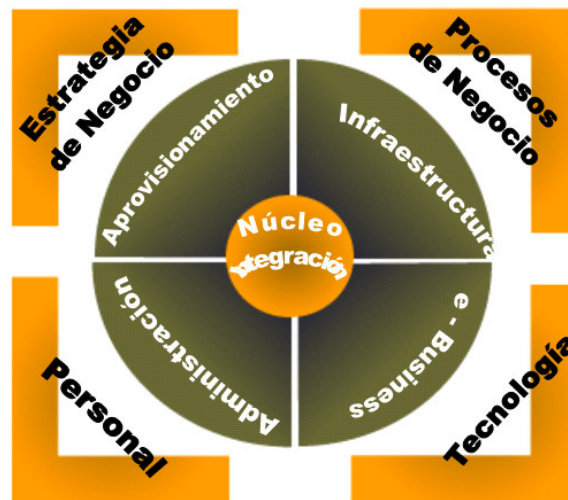


Figura 35.- Organización Conceptual SOA

Para que una arquitectura de estas proporciones pueda funcionar correctamente, se debe proveer de información útil, la cual se encuentra almacenada en grandes cantidades. Uno de los grandes problemas es decidir ¿Qué información almacenar?, ¿Cuál se considera útil? Y ¿Cuál no?. Para lograr esto se ocupan los almacenes de datos los cuales se revisarán en el siguiente tema.

## 4.6 Almacenes de Datos

Los almacenes de datos son el centro de atención para las grandes empresas de hoy en día, ya que constituyen uno de los soportes fundamentales para el proceso de toma de decisiones gerenciales; de ahí la importancia de que la información guardada en ellos sea confiable y con calidad. Uno de los procesos en la construcción de éstos y que contribuye a lograr este objetivo es la limpieza de datos.

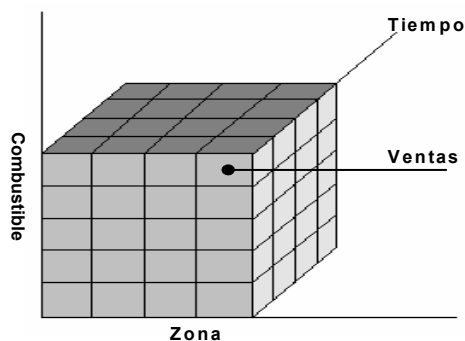
Desde un inicio, las bases de datos se convirtieron en una herramienta fundamental de control y manejo de las operaciones comerciales. Fue así como en pocos años, en grandes empresas y negocios, existía un considerable número de información almacenada en diferentes fuentes de datos y éstas ya habían alcanzado un tamaño considerablemente grande. Con esta gran acumulación de información, los directivos de tales empresas y negocios se dieron cuenta que ésta podría tener un fin útil, al estar reflejada la mayoría de sus operaciones comerciales durante los llamados ciclos de negocios propios del mercado.

Por tanto, pensaron que lo ideal sería unificar las diferentes fuentes de información de las cuales disponían, en un único lugar, al que sólo se le incorporaría información relevante, sobre la base de una estructura organizada, integrada, lógica, dinámica y de fácil explotación. La respuesta a esto fueron los Almacenes de Datos o DataWarehouse (DW).

Existen herramientas que pueden extraer información útil de un almacén de datos, una de ellas es OLAP.

### Explotación de un almacén de datos: Herramientas OLAP

Las herramientas OLAP son utilizadas para explotar las estructuras multidimensionales definidas en la base de datos. Para poder entender mejor como es que se forman cubos de información, se verán los datos dentro de una base de datos como un plano cartesiano, donde cada uno de los ejes representa una dimensión, y la intersección entre estos valores representa un hecho o medida como en la figura 36.



---

Figura 36.- Cubo de Información

Es importante hacer notar que una dimensión que debe estar siempre presente en todo modelo dimensional, es la dimensión Tiempo, ya que mediante ésta se pueden hacer comparaciones de distintos periodos, así como efectuar drill-down y Slice and Dice.

Algunas variantes de OLAP son las siguientes:

**DOLAP (Desk OLAP).** Son herramientas que generan el análisis en el cliente y no cuentan con un servidor que efectúe la función de motor de análisis multidimensional.

**ROLAP (Relational OLAP).** Tiene la misma filosofía que los DOLAP, pero sí cuentan con un motor de análisis en el servidor, con lo que se logra un mejor desempeño en la carga de datos.

**MOLAP (Multidimensional OLAP).** Tiene la característica que generan cubos de información de manera predefinida y sólo se actualiza de manera explícita. La ventaja es que provee rapidez en la manipulación de cubos, pero la desventaja es que tiene una rigidez al intentar modificar la estructura de un cubo.

**HOLAP (Hybrid OLAP).** Combinan los conceptos de ROLAP y MOLAP permitiendo flexibilidad al actualizar información de métricas con cada análisis.

Un proceso más que nos ayuda a extraer información útil entre cientos o miles de datos es la Minería de Datos (Datamining) la cual se revisará en el siguiente punto.

## 4.7 Minería de Datos (Datamining)

Datamining es el proceso que descubre información de interés, como patrones, asociaciones, cambios, anomalías y estructuras significativas, a partir de grandes volúmenes de datos almacenados en repositorios como bases de datos operacionales, Datawarehouse, etc.

Se le da el término de Minería (Mining), porque se basa en las operaciones mineras que deben realizarse para explotar en lugares vastos y profundos con el único fin de encontrar minerales valiosos, en este caso información.

Datamining nace con la necesidad de explotar el gran volumen de información que se tenían en las empresas. Datamining persigue dos objetivos fundamentales: el descubrimiento de reglas y la predicción del comportamiento.

**Motivación.** El Datamining nace del fenómeno del almacenamiento masivo de información en las bases de datos corporativas, aunado a que no se tenían herramientas tecnológicas que pudieran extraer conocimiento, además de interpretarlos.

Algunos factores que motivaron el surgimiento del Datamining fueron los siguientes:

- La mayor parte de la información es histórica.
- La información histórica es usada para predecir.
- Las decisiones de las empresas son tomadas en base a información histórica.
- La gran cantidad de variables que inciden en una decisión es muy grande.
- Los altos ejecutivos no pueden analizar gran cantidad de variables para tomar decisiones.

Para que un Datamining pueda tener éxito se debe tomar en cuenta que es necesario contar con bases de datos extensas, con gran volumen de datos.

### Relación de Datamining y otras disciplinas

Datamining se auxilia de varias disciplinas para llevar a cabo sus tareas principales y forma parte del proceso de KDD (Knowledge Discovery from Databases), es importante tener claro que KDD y Datamining no son sinónimos.

Como parte del proceso de KDD, Datamining se apoya en disciplinas como la estadística, los sistemas de información, bases de datos, inteligencia artificial, aprendizaje automático, visualización de datos, computación paralela, cómputo distribuido y lenguaje natural.

De igual manera Datamining no es una extensión de los sistemas OLAP, estos últimos son una forma de presentar la información que se ha descubierto que tiene relevancia. La figura 37 nos muestra la relación existente entre un Datamining, un KDD y otras disciplinas.

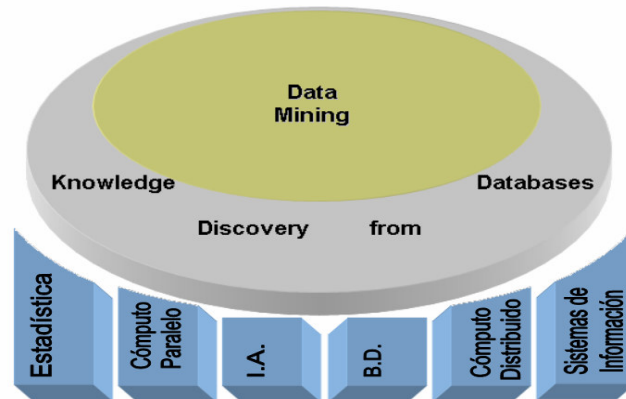


Figura 37.- Relación de un Datamining con otras disciplinas

El proceso de KDD<sup>5</sup> (Knowledge Discovery from Databases). Como se ha dicho, la minería de datos es parte de un conjunto mayor de tareas, al cual denominamos KDD.

Para KDD es importante tener claro que el conocimiento es un producto de un proceso de descubrimiento guiado por los datos. Es un proceso complejo que busca, a partir de datos o registro de una base de datos, identificar patrones válidos, novedosos, potencialmente útiles y comprensibles.

Conocer exactamente lo que se quiere buscar es suficiente para encontrarlo en nuestras bases de datos, a través de una consulta, pero, si sólo tiene una idea aproximada de lo que busca, se cae en el campo del KDD.

Generalmente un proceso de descubrimiento de conocimiento en las bases de datos contempla las siguientes fases:

- Limpieza de datos.
- Integración de datos.
- Selección de datos.
- Transformación de datos.
- Minería de datos.
- Evaluación de patrones.
- Presentación del conocimiento.

Las técnicas de minería de datos son empleadas para eficientar el rendimiento del negocio o industriales que manejan grandes volúmenes de información estructurada y almacenada en bases de datos.

<sup>5</sup> **KDD.**- Descubrimiento de información en las bases de datos.

## El problema de la Extracción Automática de Conocimiento

Las fases del desarrollo de un proyecto de minería de datos, a partir de las cuales se genera un modelo clasificador son cinco, independientemente de la técnica específica de extracción de conocimiento que se utilice. La figura 38 muestra gráficamente las fases mencionadas.

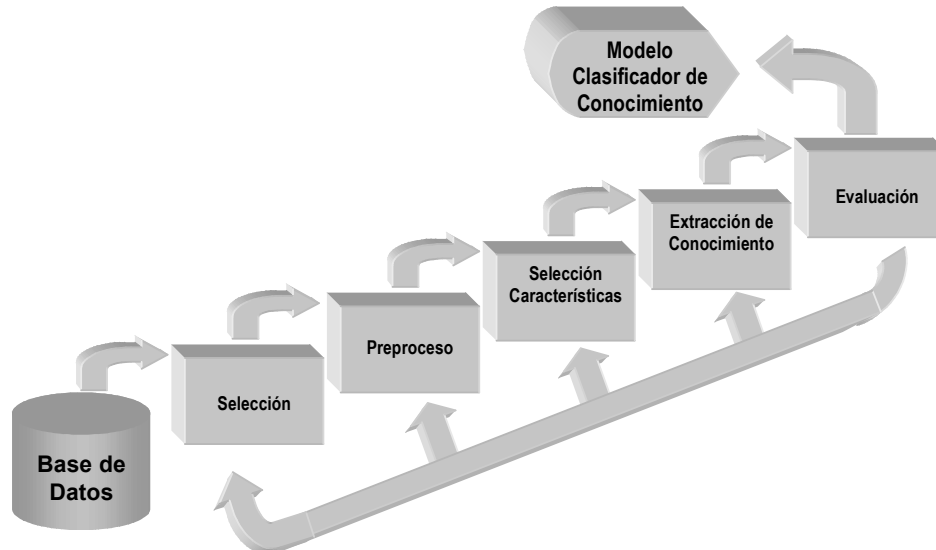


Figura 38.- Fases de Extracción de Conocimiento

**Selección.** La primera fase se enfoca en el conocimiento de los objetivos y requerimientos desde una perspectiva de negocios, a partir de la cual se selecciona una colección inicial de los datos.

**Preproceso.** En esta fase se filtran los datos con el fin de eliminar valores incorrectos, no válidos, desconocidos, etc.

**Selección y Características.** Tiene como objetivo reducir el tamaño de los datos, eligiendo las variables más significativas en el problema.

**Extracción del conocimiento.** Se analizan los datos resultando patrones de comportamiento observados en las variables del problema.

**Evaluación.** En la última fase se evalúa que los resultados obtenidos sean válidos y suficientemente satisfactorios, en caso de no ser así, se debe regresar a analizar las fases anteriores, corregir y tratar de generar nuevos modelos.

El proceso de extracción de información se vuelve diferente cuando se intenta obtener información útil de una Web, sin embargo, existe el proceso llamado Web Mining.

### Web Mining

Web Mining es el proceso de descubrimiento de conocimiento a partir de datos de la Web, ya sea cerca de los contenidos, de hábitos de navegación, etc. De acuerdo al objetivo de su búsqueda, podemos clasificar al Web Mining en tres categorías:

- **Web Content Mining.** Su objetivo es la extracción de conocimiento relacionado con el contenido de los documentos en la Web.
- **Web Structure Mining.** Su fin es el descubrimiento de un modelo partiendo de la topología de enlaces de la red.
- **Web Usage Mining.** Tiene el objetivo de la extracción de información relativa a hábitos y preferencias de los usuarios o contenidos.

Parte fundamental de la arquitectura SOA es el Datawarehouse, el cual se estudiará en el siguiente punto.

## 4.8 DataWarehouse

En esta sección se profundizará en el estudio de uno de los componentes más importantes, el Datawarehouse.

El reunir los elementos de datos apropiados desde diversas fuentes, en un ambiente integral centralizado, simplifica el problema de acceso a la información, y en consecuencia, acelera el proceso de análisis, consultas y menor tiempo de uso de la información.

Una ventaja más que tiene el Datawarehouse es que gracias a su arquitectura permite independencia de la fuente de datos y se enfoca sobre la información relevante para el negocio.

Un Datawarehouse se crea a partir de la extracción de información de diferentes bases de datos operacionales, mediante el proceso ETL. Estos datos son extraídos, transformados para eliminar inconsistencias y resumir, si es necesario, antes de ser cargadas en el Datawarehouse.

Existen cuatro características básicas que diferencian a un Datawarehouse de otro tipo de sistemas:

1. **Orientación al Tema.** Indica que la información se clasifica con base en los aspectos del negocio que son de relevancia para la empresa.
2. **Integración.** Este es muy importante ya que la información encontrada al interior debe estar siempre integrada, tanto en dominio como en significado, políticas, protocolos de acceso y plataformas.

Los elementos que deben cuidarse en la integración de la información de un Datawarehouse, se pueden clasificar en cuatro categorías principales.

- **Codificación.** Nos indica cuáles son los códigos válidos dentro del Datawarehouse.
- **Unidades de Medida.** Los sistemas pueden tener diferentes unidades de medida, para hacerlos comparables y poder integrar la información, se deben unificar las unidades de medida.



- **Convenciones de Nomenclatura.** Un reporte de alguna rea puede ser interpretado de diferentes formas según el área que lo analice, en el Datawarehouse persigue evitar este punto homologando nomenclaturas.
  - **Fuentes Múltiples.** Aquí es muy importante que el ETL no extraiga el mismo dato de diferentes fuentes, sino que debe analizar cuál es el correcto.
3. **De tiempo Variante.** Las consultas a la información en el Datawarehouse se enfocan hacia la información en distintos momentos, debe incluir los datos históricos para usarse en la identificación y evaluación de las tendencias.
  4. **No volátil.** Esta propiedad permite ver la información como fotografías que se toman en diferentes momentos y que no pueden ser alteradas, con esto se puede hacer comparaciones en diferentes tiempos, y esto hace que se tenga estable la información y se puedan hacer tendencias.

Es fundamental para un Datawarehouse los Metadatos, en ellos es donde se define toda la lógica del negocio, además de que se encuentran en un plano diferente al de los otros datos, ya que su contenido no se toma del ambiente operacional.

Los Metadatos pueden conceptuarse como:

- Un directorio que auxilia al analista a identificar los contenidos del Datawarehouse.
- Una guía para el mapeo de la transformación de los datos del ambiente operacional al del Datawarehouse.
- Una guía acerca de los algoritmos utilizados para la generación de la agregación de los datos en sus distintos niveles, desde el detalle hasta los datos completamente resumidos.
- Datos que describen datos.

En seguida se analizará cómo se forma una arquitectura de un sistema Datawarehouse:

### **Arquitectura de un sistema de Datawarehouse**

La arquitectura de un Datawarehouse puede visualizarse como una estructura de capas donde se encuentran elementos como datos, comunicaciones, procesamiento y presentación de la información.

La arquitectura de un Datawarehouse se divide en seis capas.

1. **Capa de Datos operacionales.** En esta capa se encuentra el origen de los datos que utiliza el Datawarehouse: sistemas operacionales. Es aquí donde puede volverse complejo, ya que en la mayoría de los casos hay que interactuar con diversas plataformas y la antigüedad de algunos de ellos requiere protocolos de acceso. Por otra parte, y con más frecuencia, las organizaciones grandes adquieren datos adicionales desde bases de datos externas.
2. **Capa ETL (Extracción, Transformación y Carga).** Esta capa es muy compleja pero muy importante para el Datawarehouse, es aquí donde se leen datos, se procesan, se integran y cargan, para poder hacer esto es necesario contar con un buen análisis de calidad de datos.

Puede ser tan compleja que pueden estar inmersos varios ETL's trabajando sobre diferentes bases de datos operacionales. El ETL es un proceso crítico y puede ser el que más tiempo consume en el proceso de creación del Datawarehouse.

- 3. Capa conceptual del Datawarehouse.** En esta capa se define un modelo de datos para el Datawarehouse, para ello se utiliza una técnica denominada Modelado Dimensional. Este nombre se le da a la técnica utilizada para diseñar sistemas Datawarehouse.

El modelo dimensional es una técnica que busca presentar los datos en una estructura estándar e intuitiva que permita alta eficiencia en su explotación. Este modelo está formado de tres tipos de elementos básicos: Tabla de dimensiones, Tablas de Hechos y Relaciones.

Tradicionalmente una tabla de hechos es una estructura cuya llave primaria está compuesta por una o más llaves foráneas, y que adicionalmente contiene atributos cuantificables.

En la figura 39 se puede identificar la tabla de hechos en la parte central y las dimensiones alrededor de ella.

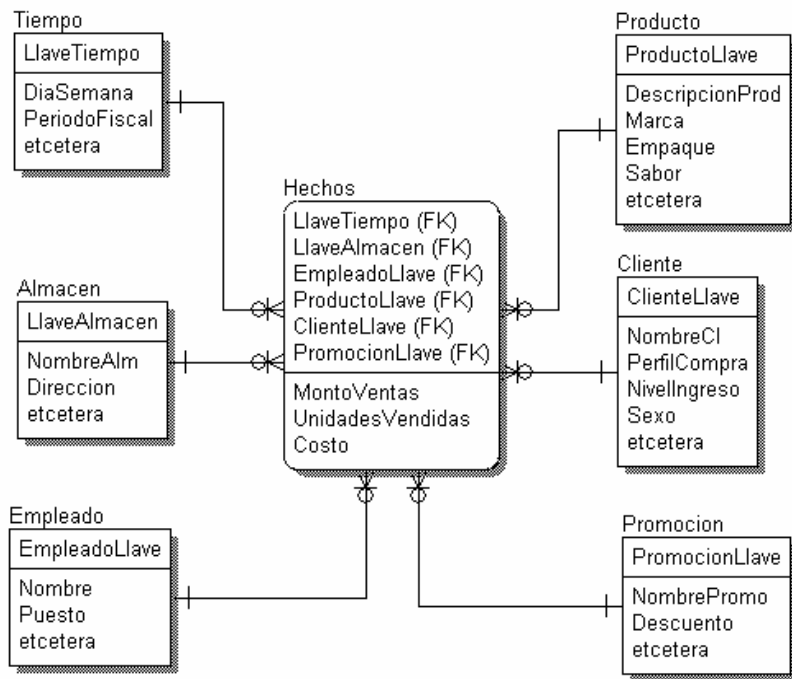


Figura 39.- Tabla de hechos y dimensiones

- 4. Capa Almacenamiento de Datos.** El acceso a los datos del Datawarehouse está íntimamente ligado con SQL (Estándar Query Language), ya que en los últimos años a llegado a ser el estándar para el intercambio de datos.

Otro elemento que ha permitido la interconexión de fuentes de datos heterogéneos, lo representa ODBC (Open Database Connectivity).

El corazón o núcleo del Datawarehouse es su base de datos, donde son vaciados los datos provenientes del ambiente operacional y está orientada para usos estratégicos.

- 5. Capa de Negocio.** Hay distintos elementos que considerar para que la implementación de un Datawarehouse en una empresa sea exitosa, y es de vital importancia considerar la capa de negocios para que esto se cumpla.

Por ejemplo, hay que tener cuidado con empresas que tienen la intención de implementar un Datawarehouse y empiezan implementando Datamarts, perdiendo el objetivo principal.

En la arquitectura, esta capa de negocios funge como la principal proveedora de insumos para las herramientas de explotación del Datawarehouse, tales como los EIS, DSS, OLAP, etc.

- 6. Capa de Presentación.** En la parte superior de la arquitectura se encuentra la capa de presentación de los datos, compuesta por los distintos tipos de herramientas utilizadas para la explotación de la capa de almacenamiento.

En la figura 40 se muestra la relación que existe entre las tecnologías y el tipo de usuario que las utiliza, generalmente se distinguen 5 tipos de tecnología de negocios que realizan la explotación de la información disponible en el Datawarehouse.

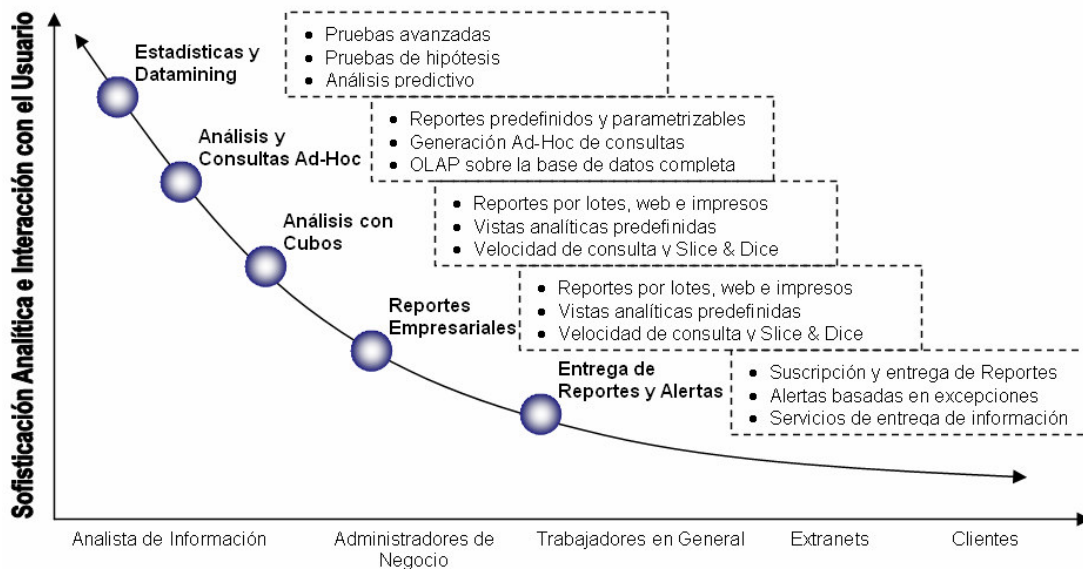


Figura 40.- Relación entre las tecnologías y el tipo de usuario que las utiliza

De la figura 40, se puede observar quién mas explotación y aprovechamiento tienen son los Analistas de Información, quienes poseen conocimientos mas profundos, en seguida los Administradores de Negocio, la media y alta dirección, los cuales son consumidores de reportes empresariales, al final se tiene la entrega de reportes y alertas, las cuales se encuentran a disposición de los trabajadores en general.

Para poder entender mejor todo lo visto, en el siguiente punto se verán algunos puntos de cómo sería aplicarlo a un proyecto real.

## 4.9 Aplicación del Método

En seguida se explicará cómo se aplica la notación en la generación de un proyecto real. En general, se divide en cuatro fases: Análisis, Diseño, Construcción y Pruebas.

### 1. Análisis

Esta fase es muy importante y se tomarán algunas actividades básicas, sin embargo, dependerá de las necesidades de cada empresa.

Lo primero que debe realizarse es la recolección y análisis de los requerimientos. Estos pueden dividirse en requerimientos de dominio específico y requerimientos arquitectónicos. Los primeros son aquellos que pertenecen al área de desempeño del usuario final, mientras que los segundos son requerimientos más relacionados con las características de construcción del sistema.

Un mecanismo arquitectónico es una solución a un problema o requerimiento frecuente, definida en distintos niveles de abstracción que permiten llegar a la especificación concreta de productos que satisfagan la necesidad identificada.

En el nivel de análisis se definen características genéricas (independientemente de la implantación) que debe cumplir el producto, mientras que en el nivel de diseño, se identifican las tecnologías que cumplen con el requerimiento analítico; por último en el nivel de instrumentación se especifican las marcas de productos concretos que cumplen con las necesidades planteadas en los niveles superiores.

Los requerimientos más importantes de esta fase, teniendo en cuenta que estos dependen completamente del negocio, son los siguientes:

- Requerimientos de dominio específico. Se refiere a la correcta identificación de las reglas de negocio.
- Identificación de métricas. Una métrica o indicador debe interpretarse como información, ya sea gráfica o textual, que nos permite conocer y medir el estado de algún proceso, situación o evento en particular.
- Validar su existencia y disponibilidad. Una vez que se identificaron las métricas es necesario validar la existencia de las fuentes donde residen los datos, ya sean internas como externas.

### 2. Diseño

En esta fase se elaboran el modelado conceptual, lógico y físico del Datawarehouse. Primeramente se diseña el esquema conceptual del Datawarehouse, identificando inicialmente, las entidades que formarán el modelo, clasificándolas en hechos y dimensiones.

Una vez que se identificaron las tablas de hechos y dimensiones, se comienza a modelar los esquemas de estrella (un hecho rodeado por n dimensiones).

La tabla de hechos tendrá una llave primaria compuesta, que se forma a partir de las llaves primarias de las dimensiones. Gráficamente se puede observar el modelo lógico en la figura 41.

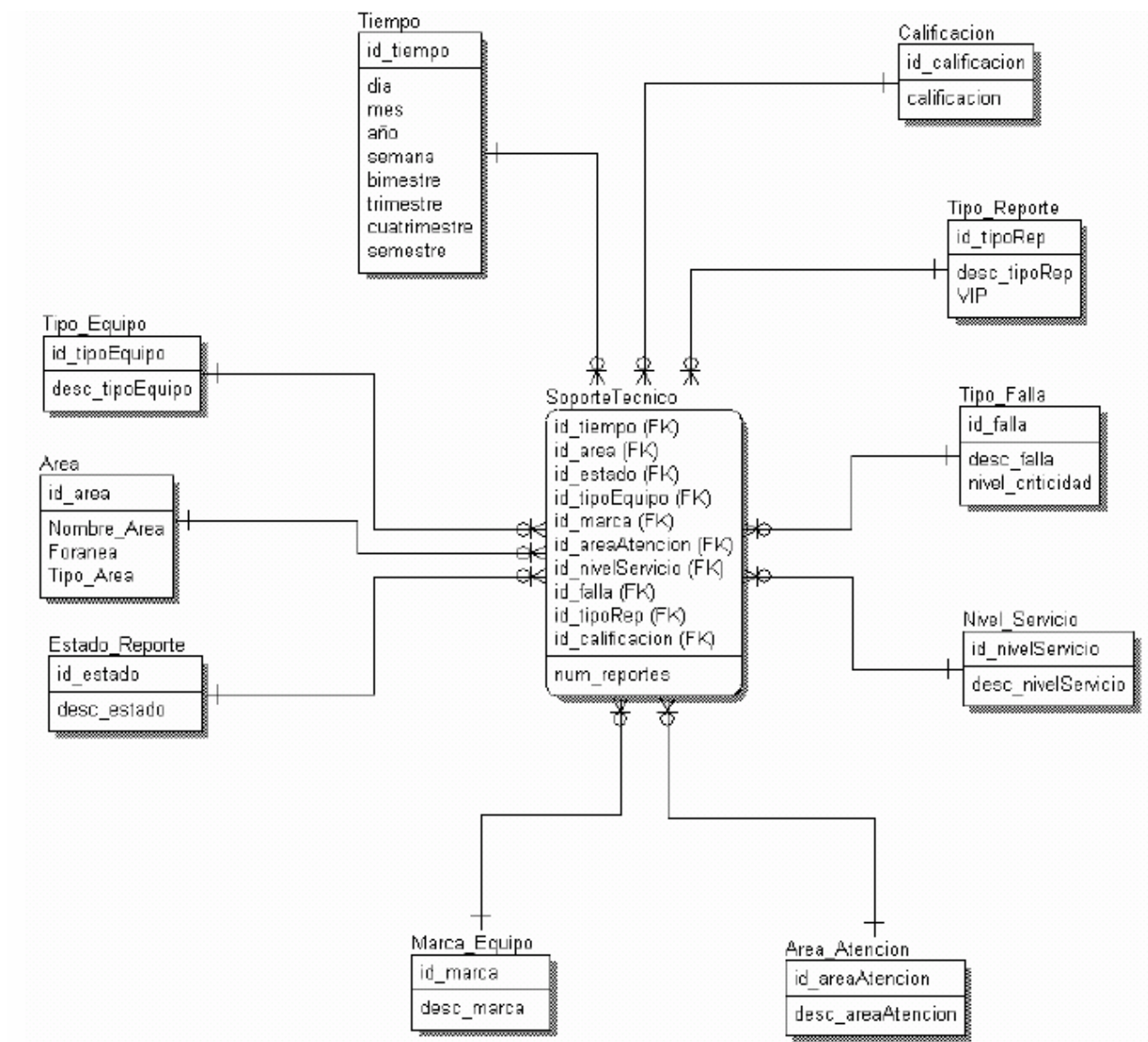


Figura 41.- Modelo Lógico

El siguiente paso es crear el modelo Físico, en el cual se agregan los tipos de datos particulares de la plataforma elegida para la instrumentación del Datawarehouse, como se muestra en la figura 42.

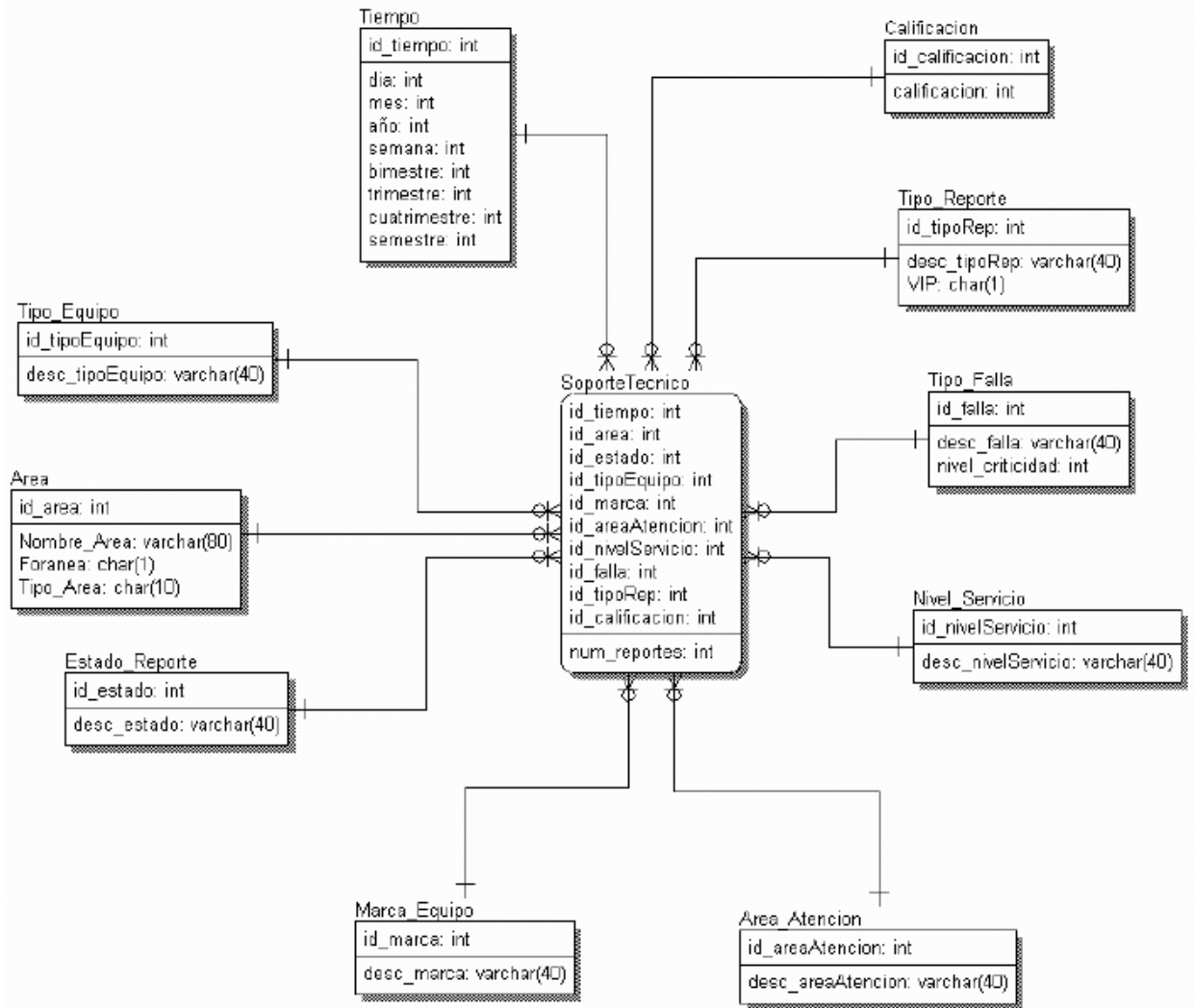


Figura 42.- Modelo Físico

Posteriormente, ya se está en posibilidad de diseñar un proceso ETL y éste a su vez, ayuda a detectar deficiencias en modelos como por ejemplo la inexistencia de métricas.

El diseño ETL se compone de siete pasos.

1. Selección de los orígenes de datos.
2. Limpieza de los orígenes.
3. Transformación de los orígenes.
4. Asociación de orígenes.
5. Selección del destino para la carga.
6. Mapeo de los atributos origen hacia los atributos destino.
7. Carga de datos.

### 3. Construcción

En esta fase se lleva a cabo la instrumentación de diseños en la plataforma seleccionada. Elegir una herramienta adecuada dependerá de un análisis, de si existe alguna en el mercado, o si se tendrá que realizar, además de tomar en cuenta la alta dirección por los costos asociados a un desarrollo como éste.

- Se inicia con la construcción de la base de datos definida en los modelos físicos del Datawarehouse.
- En seguida o de forma paralela se realiza la programación de un ETL apoyándose de herramientas especializadas o recursos como ODBC para acceder a diversas fuentes de datos diversos.
- Al final, viene la capa de usuario donde se desarrollan las interfaces que servirán para el análisis de datos y la toma de decisiones, usando los datos de prueba validados por el ETL.

### 4. Pruebas

En esta fase se validan los procesos de carga, los modelos y documentos de análisis desarrollados, además de comprobar que la información presentada cubra los requerimientos del usuario final para fundamentar su toma de decisiones.

Esta fase es iterativa con respecto a las fases mencionadas anteriormente y es importante llevar el control de pruebas con guiones o bitácoras de resultados con versiones de control y cambios.

La construcción de modelo de manera real, tal como se vio, consta de 4 fases, sin embargo existen diferentes metodologías que se encargan y se especializan en la creación de sistemas, por ejemplo RUP<sup>6</sup>, explicando todos los pasos previos que hay que tomar en cuenta, la documentación necesaria, el hardware requerido, las tecnologías existentes de programación, etc. En el siguiente capítulo se profundizará en todo lo necesario para la creación de un sistema y las tecnologías que se utilizan.

---

<sup>6</sup> **RUP.** Proceso unificado de Racional, se trata de una potente herramienta de apoyo para la administración de creación de proyectos de TI

## Capítulo 5

### Diseño de Sistemas y Tecnologías de Programación

El presente módulo tiene el objetivo de mostrar las diferentes metodologías que se usan para el desarrollo de sistemas, iniciando con proyectos pequeños hasta los más complejos que se puedan diseñar. Los primeros temas hablan de las metodologías que se usaban y que siguen sirviendo, dependiendo del tipo de sistema que se desarrolle, actualmente existe una metodología que puede amoldarse a cualquier tipo de proyecto, esta metodología se llama RUP (Proceso Unificado de Racional), el cual está basado en fases con iteraciones para que poco a poco se desarrolle el sistema detectando errores o posibles desviaciones. RUP está fuertemente relacionado con UML (Lenguaje de Modelado Unificado), el cual es gráfico y permite modelar una necesidad de un proyecto. En la segunda parte del capítulo se revisará cómo la Ingeniería de Software ha permitido mejorar y eficientar los procesos de las metodologías como RUP por ejemplo, cómo se han desarrollado herramientas que nos permiten desarrollar más confiablemente y rápidamente, documentando todos los pasos realizados, un ejemplo de ello es RAD (Desarrollo Rápido de Aplicaciones).

#### 5.1 Introducción al Análisis de Sistemas

El presente módulo tiene el objetivo de mostrar una metodología reciente y exitosa para la construcción de software, ya que a través del tiempo nos ha enseñado que en el desarrollo de software se pueden cometer diversos errores, desde obtener resultados no solicitados, por no entender el problema real desde el principio, o simplemente en la programación; esa metodología es RUP (Proceso Unificado de Racional). A continuación se revisará el ciclo de vida del software.

##### Ciclo de vida del software

El tiempo en el que se desarrolla algún sistema o software requiere dividirlo en etapas, a las cuales se les conoce como ciclo de vida del software [23], existen diversos modelos que representan esto, sin embargo, todos ellos pueden incluirse en el siguiente ciclo tradicional:

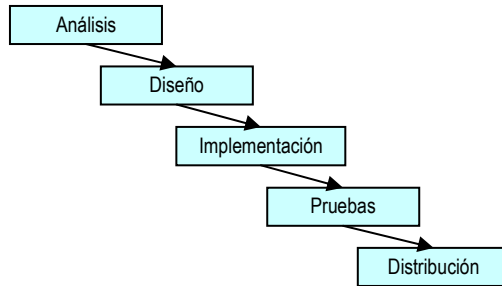
1. **Análisis y definición de necesidades.** Esta fase se documenta con los usuarios que requieren el sistema, se definen necesidades y al final se hace lo más comprensible posible para el personal de desarrollo.
2. **Diseño del Sistema y del Software.** Se determina cómo se hará (¿cómo debe ser construido el sistema?; aquí se definirán en detalle entidades y relaciones de las bases de datos, se seleccionará el lenguaje más adecuado, el Sistema Gestor de Bases de Datos a utilizar en su caso, librerías, configuraciones hardware, redes, etc.).
3. **Aplicación y pruebas de unidades.** En esta fase se realizan pruebas a los conjuntos de unidades o bloques de programa escritos en algún lenguaje de programación. Cada unidad o bloque debe cumplir con su especificación.
4. **Pruebas de Sistema.** Las unidades de programa se integran para formar un sistema completo, al cual se le harán las pruebas para verificar que cumpla con su cometido.
5. **Operación y Mantenimiento.** Se instala el programa y se pone en práctica. La actividad de mantenimiento consiste en detectar anomalías para corregirlas.



Existen diferentes metodologías que a través del tiempo se fueron desarrollando, se revisará cómo se fueron mejorando y alcanzando el desarrollo de sistemas cada vez más complejos.

### Modelo de Cascada.

El modelo en cascada establece que cada etapa debe completarse antes de que la siguiente pueda comenzar. El modelo lo muestra la figura 43.



---

Figura 43.- Modelo de Cascada Tradicional

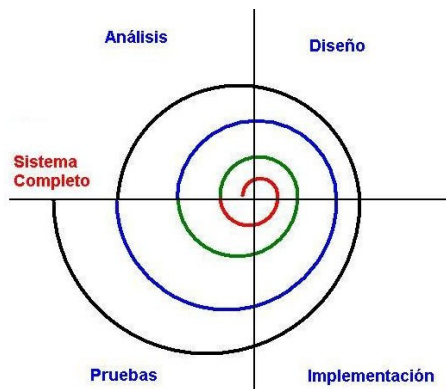
Este modelo, es conveniente para sistemas pequeños, pero se presentan problemas cuando el sistema es más grande y complejo.

Desventajas:

- Cuando el sistema es muy complejo puede volverse inmanejable, ya que no debe avanzar a la siguiente etapa hasta terminar la anterior.
- El costo de rectificar errores se incrementa exponencialmente conforme el tiempo avanza.
- Al ser un sistema grande se puede consumir mucho tiempo para su construcción y es muy complicado conservar el equipo de desarrollo.
- Se corre el riesgo de que al final de las etapas se presente un producto que no deseaba el cliente, ya que sólo se verifica en la primera etapa de Análisis.

### Modelo de Espiral

El modelo espiral propuesto por Boehm añade un elemento importante “el análisis de riesgo”. Consta de cuatro etapas, cada etapa construida a partir de la anterior y se realizan ciclos de vida cortos (iteraciones), los cuales terminan con liberación de código ejecutable. El modelo lo muestra la figura 44.



---

Figura 44.- Modelo Espiral

Ventajas:

- El equipo de desarrollo trabaja durante todo el ciclo de vida del software en periodos cortos, y no sólo en una actividad durante mucho tiempo.
- La retroalimentación con el cliente se realiza en forma temprana ya que se libera en periodos cortos código ejecutable.
- La escala y complejidad de trabajo se pueden describir en etapas tempranas.
- Los cambios que nos solicite el cliente, además de correcciones se realizan más fácilmente.
- Al liberar constantemente código, el equipo de trabajo se estimula e incentiva al equipo.

Desventajas:

- El proceso se asocia a RAD (Desarrollo Rápido de Aplicaciones) que es considerado un trabajo especializado.
- El proceso es mucho más difícil de administrar.

### Marco de Trabajo Iterativo incremental (Iterative Incremental Framework)

Este modelo es una extensión del modelo espiral solo que más formal y rigurosa. Consta de cuatro fases y las iteraciones se realizan en la fase tres. El modelo está representado en la figura 45.

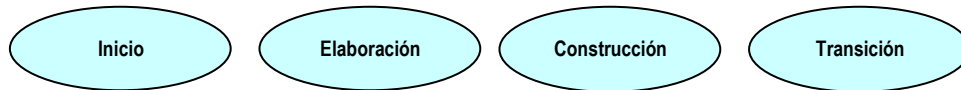


Figura 45.- Las cuatro fases del marco de trabajo iterativo incremental

- **Inicio.** Esta fase se enfoca a establecer alcances del proyecto y definir una visión para la correcta entrega de resultados.
- **Elaboración.** Consiste en analizar el problema, desarrollar detalladamente un plan de proyecto para la correcta comprensión del proyecto en general.
- **Construcción.** En esta fase se genera el producto y a diferencia de las demás no se realiza en forma secuencial, sino en iteraciones como lo hace el modelo espiral. Al final se debe tener un sistema operable. Esto está representado en la figura 46.

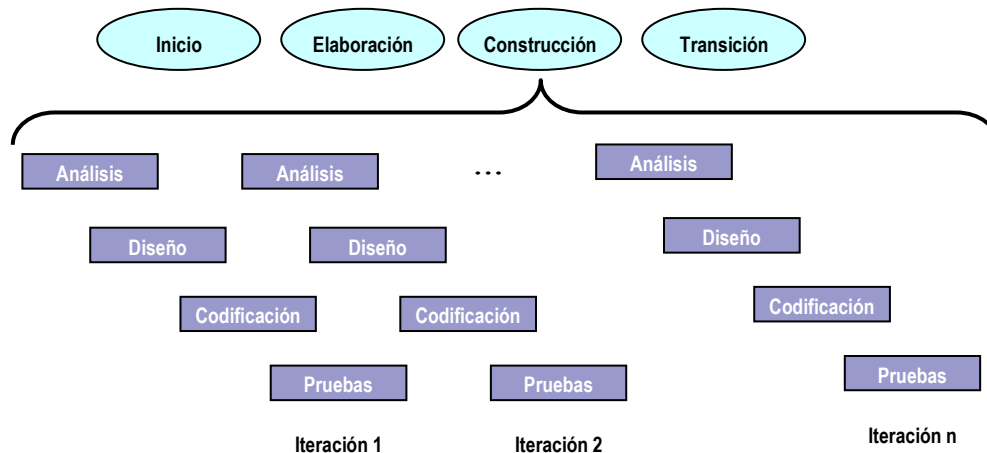


Figura 46.- La fase de construcción está formada por una serie de mini cascadas

- **Transición.** Esta etapa tiene la finalidad de trasladar el producto final a los clientes. Algunas actividades típicas de esta fase son las siguientes:
  - Liberaciones beta para que sean probadas por la comunidad de usuarios.
  - Pruebas en paralelo con el sistema antiguo para que sea sustituido.
  - Manipulación de datos (población de bases de datos, transformación de datos, etc.).
  - Capacitación a los nuevos usuarios
  - Mercadotécnica, distribución y ventas.

El número de iteraciones en la fase de construcción dependerá de qué tan grande sea el proyecto. Un proyecto grande y complejo no indica que las iteraciones sean largas, sino más bien deberá haber más iteraciones. Una simple iteración debe consumir entre dos semanas y dos meses, un tiempo mayor incrementa la complejidad.

### El proceso Unificado de Racional (RUP)

El RUP o Proceso Racional Unificado [24] (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, forman la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Sus principales características son:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software:
  - Desarrollo iterativo.
  - Administración de requisitos.
  - Uso de arquitectura basada en componentes.
  - Control de cambios.
  - Modelado visual del software.
  - Verificación de la calidad del software.

RUP Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye Artefactos y Roles. Artefactos son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc., y el rol es el papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso. RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante. RUP está fuertemente relacionado con UML, la cual es una herramienta de la cual se apoya para formar una poderosa herramienta, en el siguiente tema se analizará UML.

## 5.2 UML

El Lenguaje de Modelado Unificado UML [25] (Unified Model Language) es la herramienta gráfica que proporciona una sintaxis para describir los elementos principales (llamados artefactos en UML) usada en la descripción y construcción de software, reconocida por la industria como estándar, mientras que el Proceso Unificado Racional (Rational Unified Process) es la metodología de IBM Rational para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología.

En la década de los noventa tiene lugar un periodo conocido como “la guerra de los métodos”, habían en uso más de 50 modelos, cada uno de ellos con diferente sintaxis, y algunas semejanzas. A mediados de los noventa surgen tres métodos como los más poderosos (Método de Booch creado por Grady Booch, La técnica de modelado de objetos creado por Jim Rumbaugh y la Ingeniería de software orientada a objetos creado por Ivar Jacobson)<sup>7</sup>, los cuales convergen para formar UML.

UML funciona para cualquier tipo de proyecto, puede ser para algo pequeño y simple o para algo muy grande y complejo, es decir, se amolda a cada necesidad. Por lo tanto UML es un lenguaje genérico, en otras palabras es simplemente, una notación, una sintaxis. En seguida se revisarán algunos conceptos fundamentales para ir comprendiendo el tema.

### Conceptos principales en UML

En un proyecto de desarrollo intervienen varios participantes y cada uno de ellos necesita información referida a su función en el desarrollo de un sistema, para ello UML cuenta con diferentes diagramas que apoyan a cada uno de estos participantes con el fin de hacer mucho más fácil el manejo de la creación del proyecto. Al final cada participante podrá apoyarse, al menos, de un diagrama de UML.

Algunos participantes pueden ser:

- Analistas
- Diseñadores
- Codificadores
- Probadores (testers)
- Evaluadores de Calidad (QA)
- Documentadores
- Clientes

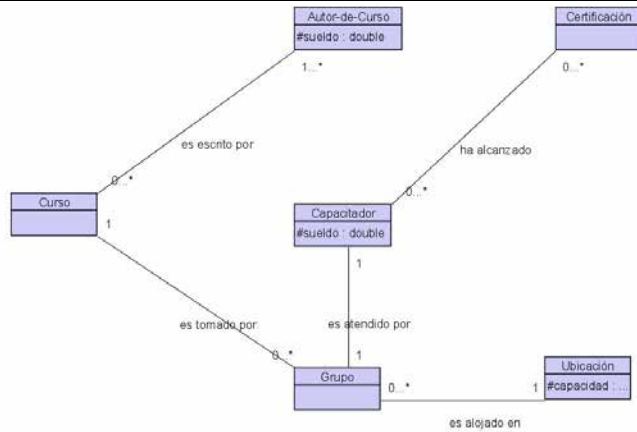
**Diagramas.** En las siguientes tablas se muestran algunos de los diagramas más utilizados, dando una definición a la izquierda y a la derecha su correspondiente diagrama.

Diagrama de Casos de Uso	
Sirve para describir las interacciones del sistema con su entorno, identificando los Actores, que representan los diferentes roles desempeñados por los usuarios del sistema, y los Casos de Uso, que corresponden a la funcionalidad que el sistema ofrece a sus usuarios, explicada desde el punto de vista de éstos.	<pre>graph LR;   Operador((Operador)) --- Arrancar([Arrancar sistema]);   Operador --- Cerrar([Cerrar sistema]);   Operador --- Producir([Producir reporte]);   Sistema[Sistema de pedidos] --- VerEstado([Ver estado-de-pedido]);</pre>

<sup>7</sup> Al equipo de Rumbaugh, Booch y Jacobson se le conoce como los “Three amigos”.

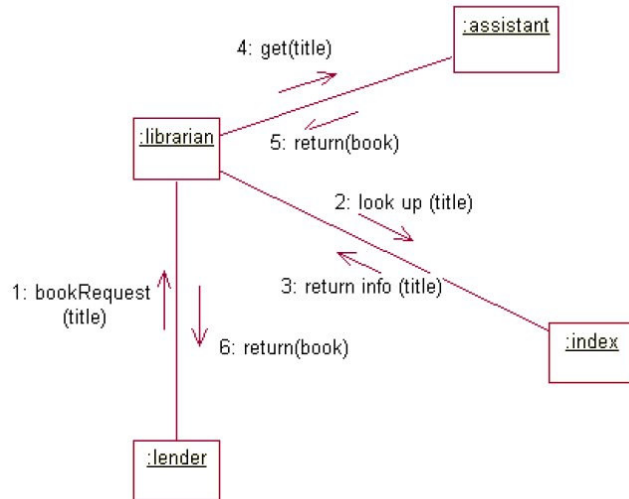
**Diagrama de Clases**

Un diagrama de clases es una colección de elementos de un modelo orientado a objetos, tales como clases, interfaces, y sus relaciones, conectados entre sí y con sus contenidos.



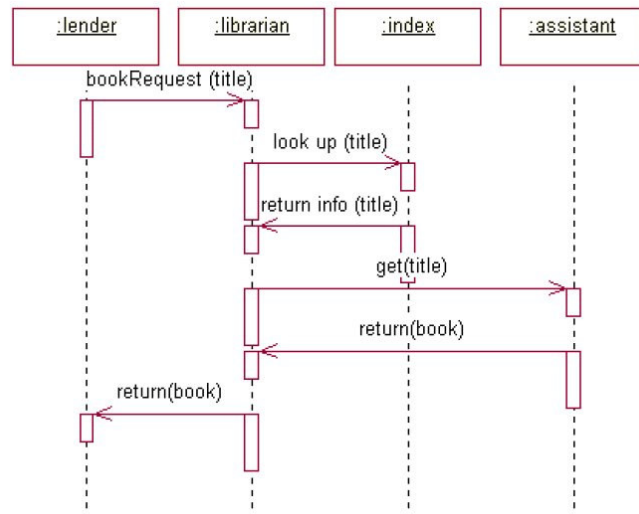
**Diagrama de Colaboración**

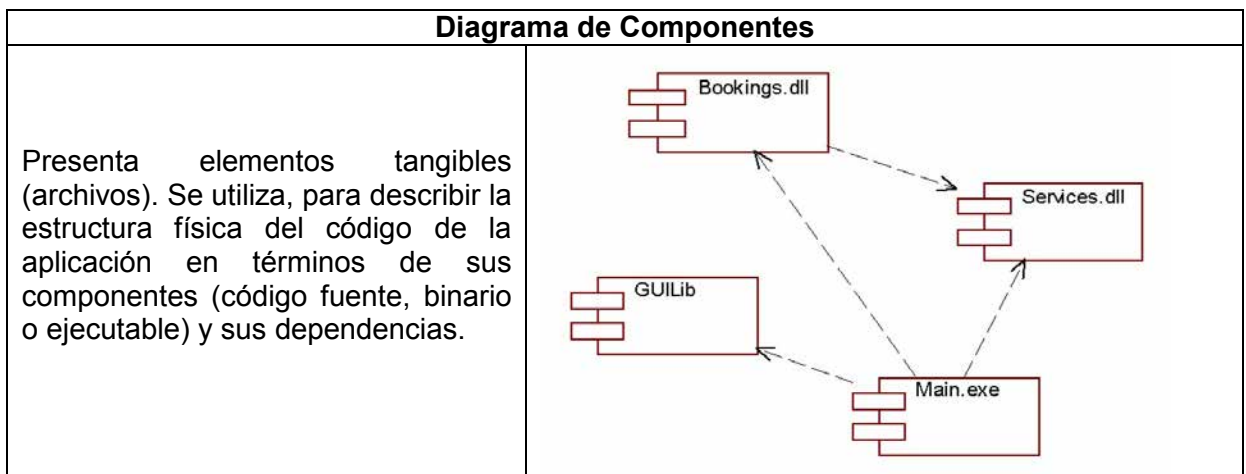
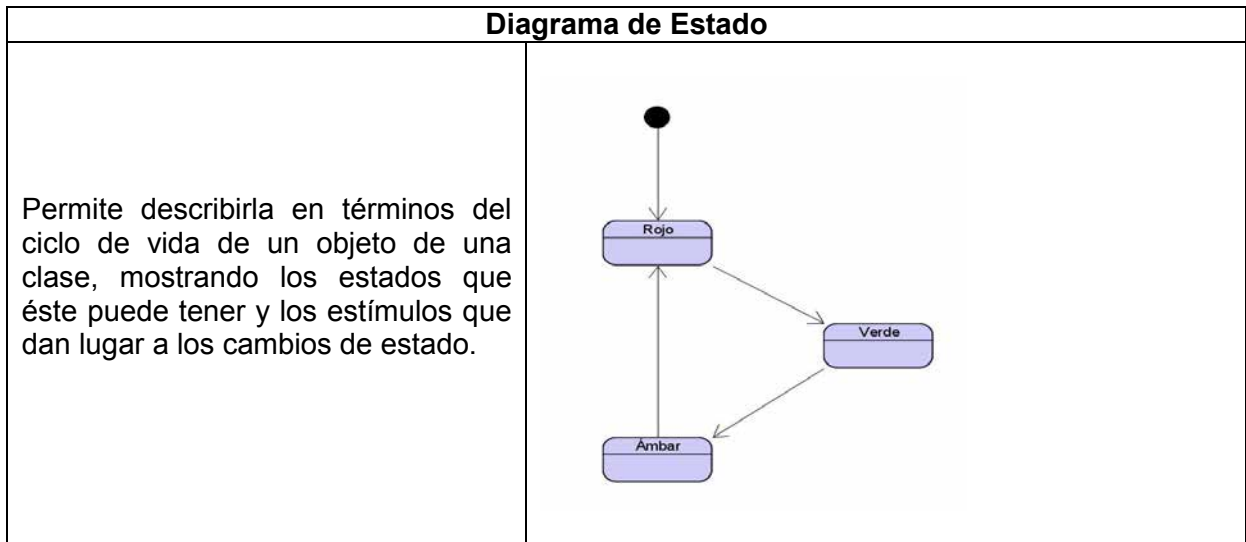
Los diagramas de colaboración muestran cómo se relacionan los objetos construidos entre sí, este diagrama es sumamente importante, ya que nos ayuda a entender con mucha facilidad como estará formado el sistema.



**Diagrama de Secuencias**

Un Diagrama de Secuencias contribuye a la descripción de la dinámica del sistema en términos de la interacción entre sus objetos respecto al tiempo.





Tal como se ha mencionado RUP y UML forman la metodología más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. En seguida se revisará en qué consisten las fases de RUP.

### 5.3 Fases de RUP

RUP divide el proceso de desarrollo en ciclos o iteraciones, teniendo un producto final al final de cada ciclo; cada ciclo se divide en fases.

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

1. Inicio (puesta en marcha)
2. Elaboración (definición, análisis, diseño)
3. Construcción (implementación)
4. Transición (fin del proyecto y puesta en producción)

La figura 47 muestra un panorama de cómo funciona RUP.

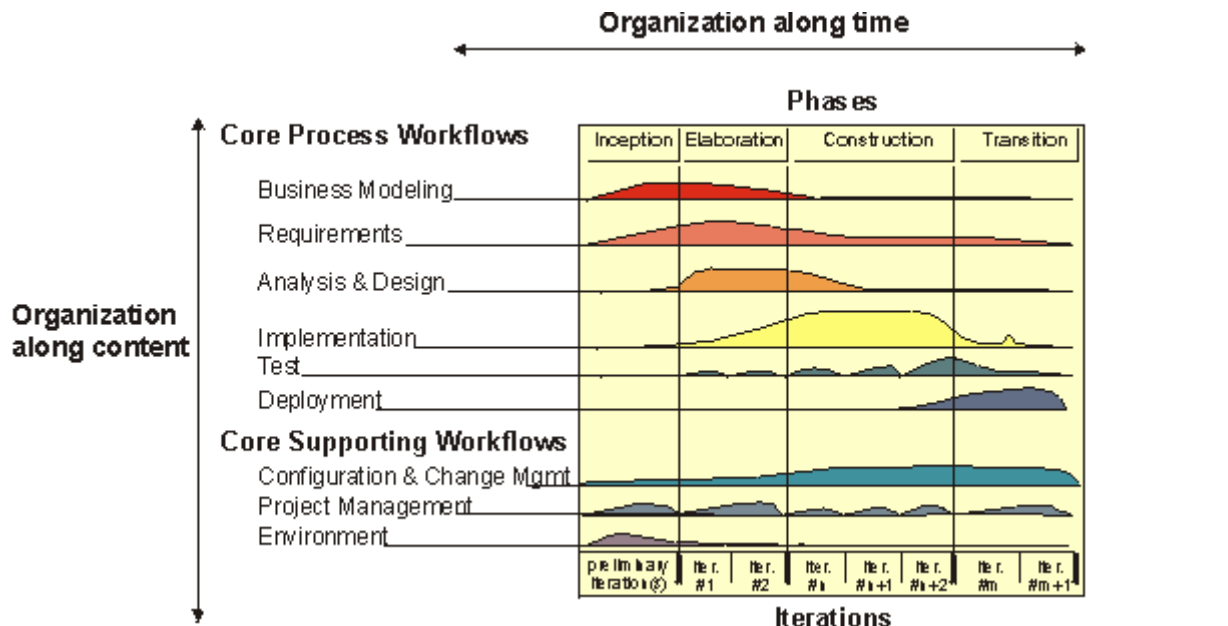


Figura 47.- Vista General de RUP

RUP define nueve actividades a realizar en cada fase del proyecto

1. Modelado del negocio
2. Análisis de requisitos
3. Análisis y diseño
4. Implementación
5. Test
6. Distribución
7. Gestión de configuración y cambios
8. Gestión del proyecto
9. Gestión del entorno

En seguida se revisará cada fase de RUP indicando cuál es la función de cada una de ellas, que se realiza y qué diagramas de UML serían de apoyo.

1. **Fase de Inicio.** En esta fase se hace un plan de fases, se define el alcance del proyecto, se identifican los principales casos de uso, se identifican los riesgos y se estima un costo total del proyecto.
2. **Fase de Elaboración.** En esta fase se hace un plan de proyecto, el equipo de desarrollo pretenderá explorar el problema en detalle entendiendo los requerimientos del cliente, se completan los casos de uso y se eliminan los riesgos.

En esta fase se desarrollan dos modelos de UML que ayudarán a entender el problema con un todo: el modelo de casos de uso y el modelo conceptual.

➤ **Modelo de casos de Uso**

El diagrama de casos de uso representa la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en cómo los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

Elementos:

- **Actor:**



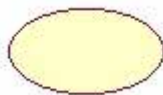
Cliente

La notación en UML es muy simple, sólo se coloca el nombre del que representará al actor, en este caso un Cliente.

Un actor es algo o alguien que puede iniciar un caso de uso, es decir, puede ser algo más que personas como algún elemento externo al sistema o algo como la fecha u hora.

Los actores sólo se pueden conectar a los casos de uso a través de asociaciones. Una asociación entre un actor y un caso de uso indican que el actor y el caso de uso se comunican entre sí, y cada uno puede enviar y recibir mensajes.

- **Caso de Uso:**



Retirar dinero

La notación de un caso de uso esta generalmente formada mediante la combinación verbo/nombre, por ejemplo en una escuela: "editar horario", "pedir listas", etc.

Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso. Un caso de uso describe que hace un sistema pero no especifica cómo lo hace.

- **Relaciones:** Algunas relaciones son las siguientes:

- **Asociación** 

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

- **Dependencia o Instanciación** 

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

- **Generalización** 

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>).

Este tipo de relación esta orientado exclusivamente para casos de uso (y no para actores).



Una labor básica es hacer una *búsqueda de casos de uso*, y esto se vuelve una tarea complicada al identificar todos los casos de uso existentes en el sistema, sin embargo, un papel sumamente importante lo juegan las personas que usarán el sistema, de esta manera se realizan entrevistas. Una estrategia más a seguir es realizar talleres conjuntos de planeación de requerimientos (Join Requirements Planning Workshop o JRP) en donde se involucran un grupo de personas interesadas en el desarrollo, llevando la siguiente dinámica:

1. Primero se realiza una tormenta de ideas de todos los actores posibles.
2. Segundo se realiza una tormenta de ideas de todos los posibles casos de uso.
3. Una vez que la tormenta de ideas se completó se debe justificar cada caso de uso haciendo una descripción simple.
4. Registrar todo el modelo.

Una vez detectados los posibles casos de uso para el desarrollo del sistema, es aconsejable proyectarlos en diagramas de Casos de Uso de UML para mejor comprensión de las relaciones de unos con otros. Una herramienta más que ayuda al mejor entendimiento del problema es el modelo conceptual, el cual se desarrolla dentro de la fase de elaboración.

### ➤ **Modelo Conceptual**

Este modelo ayuda a entender el problema con mucho más detalle junto con el cliente en una primera etapa de análisis. Para este fin nuevamente UML nos ofrece un diagrama para expresar esto, el modelo es el diagrama de clases.

Dentro de la fase de análisis, el diagrama de clases ayuda a comprender mejor el sistema, sólo que hay que considerar que no contendrá decisiones de diseño, sólo muestra una idea de cómo funciona el diseño, para no confundir términos se dirá que hay un diagrama de clases de análisis y un diagrama de clases de diseño, puesto que en este momento se hace referencia al primero.

El objetivo para desarrollar el modelo conceptual es identificar los conceptos que el cliente reconoce para el desarrollo del sistema, por ejemplo:

- Pedido, es un sistema de compras
- Empleado, es un sistema de nómina

Un ejemplo malo es el siguiente:

- ProcesoDepuracionPedidos, proceso que borra regularmente los pedidos viejos del sistema.

Este ejemplo es malo, ya que está enfocado al diseño, es decir, está enfocado a la solución y no al problema. La regla principal es la siguiente: Si el cliente no entiende el concepto, probablemente no es un concepto.

Para la búsqueda de los posibles conceptos se recomienda lo mismo que en el caso de la búsqueda de casos de uso, es decir, un taller donde se involucre a un grupo de personas interesadas en el sistema donde se realice una tormenta de ideas y posteriormente se analice cada concepto.

Enseguida se analizará cómo se representa el modelo conceptual en UML, describiendo sus elementos principales:

### El modelo conceptual en UML

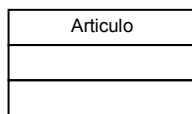
Tal como se mencionó, las características de un diagrama de clases de UML serán de gran utilidad para entender el modelo conceptual en la fase de elaboración de RUP.

El modelo se representa por medio de una caja rectangular y está formado por los siguientes elementos:

- Nombre del concepto
- Atributos del concepto
- Comportamiento del concepto
- Asociaciones

Elementos:

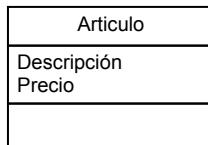
- **Nombre del concepto:**



La caja está dividida en tres partes, en la parte superior se coloca el nombre del concepto, por convención la primera letra mayúscula.

Es aquí donde aplica la siguiente regla en la búsqueda de conceptos con los clientes: Si el cliente no entiende el concepto, probablemente no es un concepto.

- **Atributos del concepto:**

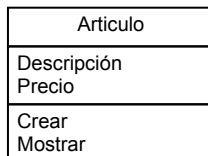


En la parte media se colocan los atributos del concepto.

De igual manera en la búsqueda de atributos puede haber confusiones, en sí debe ser un atributo o no, en este caso aplicamos la siguiente regla:

Si existe duda sobre si debe ser atributo o concepto, entonces hay que convertirlo en concepto.

- **Comportamiento del concepto:**



En la parte inferior se coloca el comportamiento del concepto.

Esta parte generalmente queda pendiente hasta la fase de construcción, ya que en estos momentos es complicado saber que hará exactamente el concepto.

• **Asociaciones:**

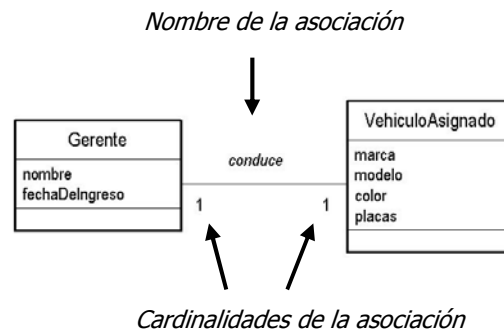


Figura 48.- Asociaciones

En UML se expresa una asociación conectando los rectángulos por medio de una línea y hay dos cosas importantes a considerar: primero, la asociación tiene un nombre, en este caso “conduce”, segundo, hay números en los extremos de la asociación, que describen la cardinalidad de la asociación. La figura 48 nos muestra una asociación.

En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y algunas se muestran en la figura 49.

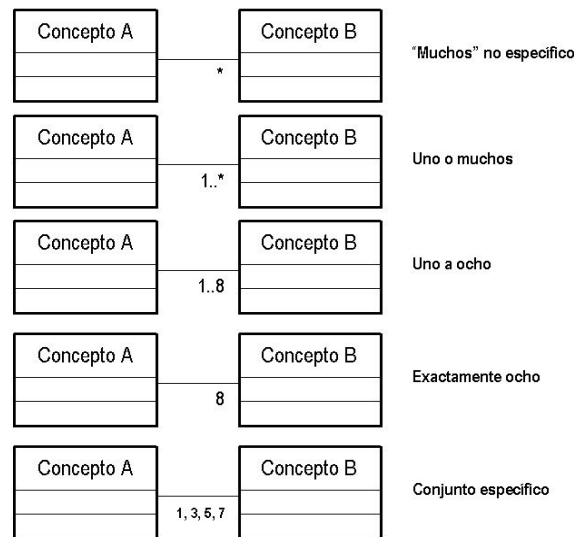


Figura 49.- Tipos de Asociaciones

**3. Fase de Construcción.** En la fase de construcción se lleva a cabo una serie iterativa de pequeñas cascadas, desarrollando en cada iteración un pequeño número de casos de uso revisando al final el avance y los límites de tiempo de cada iteración.

Una etapa en cascada producirá documentos y modelos de UML.

- En la etapa de análisis, se producirán algunos casos de uso, extendidos (o completos).
- En la etapa de diseño, se producen diagramas de clases, modelos de iteración y diagramas de estado.
- En la etapa de codificación, se produce código ejecutable y de unidades.

Al final de cada iteración idealmente se debe tener un sistema ejecutable (por supuesto, limitado). En la figura 50, muestra un panorama general de cómo está formada la fase de construcción en RUP con pequeñas cascadas.

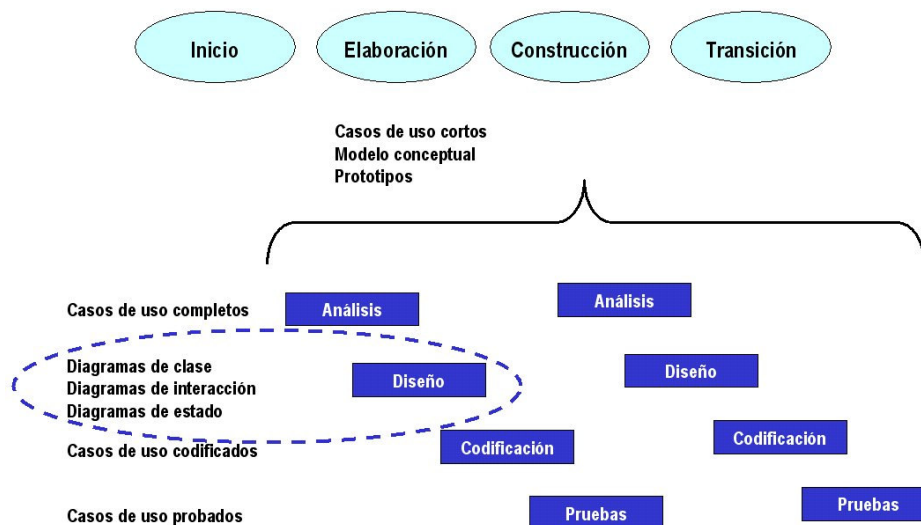


Figura 50.- Iteraciones en Fase de Construcción

Algunas características de la fase de construcción son las siguientes:

- En esta fase todos los componentes restantes se desarrollan e incorporan al producto.
- Todo es probado en profundidad.
- El énfasis está en la producción eficiente y no en la creación intelectual.
- Puede hacerse construcción en paralelo, pero esto exige una planificación detallada y una arquitectura muy estable.

**4. Fase de Transición.** Algunos objetivos que se pretenden cubrir en esta fase son los siguientes:

- Obtener autosuficiencia de parte de los usuarios.
- Concordancia en los logros del producto de parte de las personas involucradas.
- Lograr el consenso cuanto antes para liberar el producto al mercado.

Algunas características que se deben cumplir en esta fase son las siguientes:

- El objetivo es traspasar el software desarrollado a la comunidad de usuarios.
- Una vez instalado surgirán nuevos elementos que implicarán nuevos desarrollos (ciclos).

Al final cuando se tenga instalado el sistema se deben realizar las siguientes tareas:

- Pruebas Beta para validar el producto con las expectativas del cliente
- Ejecución paralela con sistemas antiguos
- Conversión de datos

- Entrenamiento de usuarios
- Distribuir el producto

RUP, por lo tanto, es una poderosa herramienta que permite desarrollar sistemas, tanto pequeños como muy complejos, ya que es muy adaptable. Sin embargo, siguiendo enlazando los temas para poder desarrollar software de calidad es necesario que se revise qué lenguaje de programación se utilizará y cuál paradigma de programación se usará, es por ello que en el siguiente tema se describirán los lenguajes de programación.

### 5.4 Lenguajes de Programación

Un lenguaje de programación [26] es una notación para escribir programas, a través de los cuales se puede realizar una comunicación con el hardware y dar así, las órdenes adecuadas para la realización de un determinado proceso. Un lenguaje está definido por una gramática o conjunto de reglas que se aplican a un alfabeto constituido por el conjunto de símbolos utilizados. Los distintos niveles de programación existentes permiten acceder al hardware, de tal forma que según se utilice un nivel u otro, así se tendrá que utilizar un determinado lenguaje ligado a sus correspondientes traductores.

#### Historia de los lenguajes de programación

Tras el desarrollo de las primeras computadoras surgió la necesidad de programarlas para que realizaran las tareas deseadas. Los lenguajes más primitivos fueron los denominados lenguajes máquina. Como el hardware se desarrollaba antes que el software, estos lenguajes se basaban en el hardware, con lo que cada máquina tenía su propio lenguaje y por ello la programación era un trabajo costoso, válido sólo para esa máquina en concreto.

La evolución de los lenguajes de programación se puede dividir en 5 etapas o generaciones.

- Primera generación: lenguaje máquina.
- Segunda generación: se crearon los primeros lenguajes ensambladores.
- Tercera generación: se crean los primeros lenguajes de alto nivel. Ejemplo: C, Pascal, Cobol.
- Cuarta generación. Son los lenguajes capaces de generar código por sí solos, son los llamados RAD, con lo cuales se pueden realizar aplicaciones sin ser un experto en el lenguaje. Aquí también se encuentran los lenguajes orientados a objetos, haciendo posible la reutilización de partes del código para otros programas. Ej. Visual.
- Quinta generación: aquí se encuentran los lenguajes orientados a la inteligencia artificial. Estos lenguajes todavía están poco desarrollados. Ej. LISP

**¿Qué es un paradigma de programación?** Un paradigma de programación [27] es un modelo básico de diseño y desarrollo de programas, que permite producir programas con unas directrices específicas. Un paradigma de programación es una colección de modelos conceptuales que juntos modelan el proceso de diseño y determinan, al final, la estructura de un programa.

Un paradigma no es mejor que otro, sino que cada uno tiene ventajas y desventajas. También hay situaciones donde un paradigma resulta más apropiado que otro.

Algunos ejemplos de paradigmas son los siguientes:

- **Paradigma de programación imperativa.** Describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea.

El paradigma imperativo debe su nombre al papel dominante que desempeñan las sentencias imperativas. El paradigma imperativo es considerado el más común y está representado, por ejemplo, por el C o por BASIC.

- **Paradigma de programación estructurada**

El paradigma de programación estructurada se basa en estructuras modulares, con fuerte cohesión en el módulo y bajo acoplamiento entre ellos. Es una forma de escribir programación de ordenador de forma *clara*, para ello utiliza únicamente tres estructuras: secuencial, selectiva e iterativa; siendo innecesario y no permitiéndose el uso de la instrucción o instrucciones de transferencia incondicional (GOTO).

Entre los lenguajes que pueden considerarse especializados en este paradigma son: lenguaje C y pascal.

- **Paradigma de programación funcional**

El paradigma funcional está basado en el modelo matemático de composición funcional. En este modelo, el resultado de un cálculo es la entrada del siguiente, y así sucesivamente hasta que una composición produce el valor deseado.

Los programas escritos en un lenguaje funcional están constituidos únicamente por definiciones de funciones, entendiendo éstas no como subprogramas clásicos de un lenguaje imperativo, sino como funciones puramente matemáticas. El paradigma funcional está representado por la familia de lenguajes LISP, en particular Scheme.

- **Paradigma de programación lógica**

La programación lógica consiste en la aplicación del conocimiento sobre lógica para el diseño de lenguajes de programación; no debe confundirse con la disciplina de la lógica computacional.

El lenguaje de programación lógica por excelencia es Prolog, que cuenta con diversas variantes. La programación lógica encuentra su hábitat natural en aplicaciones de inteligencia artificial.

- **Paradigma de programación orientada a objetos**

La Programación Orientada a Objetos (POO u OOP por sus siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan *estado* (es decir, datos), *comportamiento* (esto es, procedimientos o métodos) e *identidad* (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer programas y módulos más fáciles de escribir, mantener y reutilizar.

El paradigma orientado a objetos. Un lenguaje completamente orientado a objetos es Smalltalk.

- **Paradigma de programación orientada por capas**

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles, y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Sin embargo, el diseño de Software es cada vez más complejo y complicado de realizar, es por ello que nace la Ingeniería de Software, la cual brinda soporte y mantenimiento para cualquier sistema, enseguida se estudiará el tema al respecto.

### 5.5 Ingeniería de software y herramientas

La ingeniería de software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de 3 elementos clave: métodos o modelos, herramientas y procedimientos, estos facilitan al gestor a controlar el proceso de desarrollo de software y provee las bases para construir software de alta calidad.

1. **Los métodos de la ingeniería de software.** Suministran el cómo construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requerimientos del sistema y del software, diseño de procedimientos algorítmicos, codificación, prueba y mantenimiento. Los métodos de la ingeniería de software introducen frecuentemente una notación especial orientada al lenguaje o gráfica y a un conjunto de criterios para la calidad del software.

Existen métodos que se adecuan mejor dependiendo del tamaño y complejidad del sistema. A estos métodos se les conoce como modelos de ciclo de vida de los sistemas los cuales ya se revisaron en el apartado 5.1 Introducción al Análisis de Sistemas. Algunos de ellos son los siguientes:

- Modelos de cascada
- Modelos de Espiral
- Modelo de trabajo iterativo incremental
- El proceso Unificado de Racional (RUP)

2. **Las herramientas de ingeniería de software.** Suministran un soporte automático o semiautomático para los métodos. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software llamado ingeniería de software asistido por computadora, por mencionar alguna de estas herramientas existen las herramientas CASE. Las herramientas son cómo voy a aplicar los métodos para tener un desarrollo.

Las herramientas de ingeniería de software son los métodos necesarios para desarrollar el sistema.

Un ejemplo es UML (Lenguaje de Modelado Unificado), el cual ya se estudió en el apartado 5.2 UML.

3. **Los procedimientos de la ingeniería de software.** Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas que se requieren y los controles que ayuden asegurar la calidad y coordinar los cambios y las guías que facilitan a los gestores de software establecer el desarrollo. Un ejemplo que puede apoyar en la gestión del desarrollo de proyectos es Project de Microsoft.

Sin embargo, existen métodos para desarrollar software rápidamente con resultados buenos, los cuales aplican para situaciones que requieren rapidez.

## 5.6 Modelo de Desarrollo Rápido de Aplicaciones (RAD)

Como definición de RAD [28] (Rapid Application Development) se puede decir que se trata de un proceso de desarrollo de software que permite construir sistemas utilizables en poco tiempo, frecuentemente con algunas concesiones.

Algunos problemas atendidos por RAD son los siguientes:

- Con los métodos convencionales pasa un gran lapso de tiempo antes de que el cliente vea resultados.
- Con los métodos convencionales el desarrollo llega a tardar tanto que para cuando el sistema está listo para utilizarse los procesos del cliente han cambiado radicalmente.
- Con los métodos convencionales no hay nada hasta que el 100% del proceso de desarrollo se ha realizado, entonces se entrega el 100% del software.

Se pueden enumerar algunas características principales y desmenuzar cada una de ellas:

1. Equipos Híbridos
  2. Herramientas Especializadas
  3. "Timeboxing"
  4. Prototipos Iterativos y Evolutivos.
1. **Equipos Híbridos.** Equipos compuestos por alrededor de seis personas, incluyendo desarrolladores y usuarios de tiempo completo del sistema, así como aquellas personas involucradas con los requisitos. Los desarrolladores de RAD deben ser: analistas, diseñadores y programadores en uno.
  2. **Herramientas Especializadas.** Estas herramientas se caracterizan por estar orientadas a:
    - Desarrollo "visual"
    - Creación de prototipos falsos (simulación pura)
    - Creación de prototipos funcionales
    - Múltiples lenguajes
    - Calendario grupal
    - Componentes reutilizables
    - Control de versiones



3. **"Timeboxing"**. Las funciones secundarias son eliminadas como sea necesario para cumplir con el calendario a como dé lugar.
4. **Prototipos Iterativos y Evolutivos**. Trabajan de la siguiente manera:
  - Reunión JAD (Joint Application Development):
    - Se reúnen los usuarios finales y los desarrolladores.
    - Lluvia de ideas para obtener un borrador inicial de los requisitos.
  - Iterar hasta acabar:
    - Los desarrolladores construyen y depuran el prototipo basado en los requisitos actuales.
    - Los diseñadores revisan el prototipo.
    - Los clientes prueban el prototipo, depuran los requisitos.
    - Los clientes y desarrolladores se reúnen para revisar juntos el producto, refinar los requisitos y generar solicitudes de cambios.
    - Los cambios para los que no hay tiempo no se realizan. Los requisitos secundarios se eliminan si es necesario para cumplir el calendario.

#### Ventajas de RAD

1. Comprar puede ahorrar dinero en comparación con construir.
2. Los entregables pueden ser fácilmente trasladados a otra plataforma.
3. El desarrollo se realiza a un nivel de abstracción mayor.
4. Visibilidad temprana.
5. Mayor flexibilidad.
6. Menor codificación manual.
7. Mayor participación de los usuarios.
8. Posiblemente menos fallas.
9. Posiblemente menor costo.
10. Ciclos de desarrollo más pequeños.
11. Interfaz gráfica estándar.

#### Desventajas de RAD

1. Comprar puede ser más caro que construir.
2. Costo de herramientas integradas y equipo necesario.
3. Progreso más difícil de medir.
4. Menos eficiente.
5. Menor precisión científica.
6. Riesgo de revertirse a las prácticas sin control de antaño.
7. Más fallas (por síndrome de "codificar a lo bestia").
8. Prototipos pueden no escalar, un problema mayúsculo.
9. Funciones reducidas (por "timeboxing").
10. Dependencia en componentes de terceros: funcionalidad de más o de menos, problemas legales.
11. Interfaz gráfica estándar.
12. Funciones no deseadas.

Tal como se revisó, el desarrollo de software ha generado que se desarrollen diversas herramientas para obtener calidad, eficacia, y sobre todo no dejando de lado el tema de seguridad. Existe otra disciplina que ofrece una diferente forma de programar y que actualmente ha sido utilizado por la mayoría de las aplicaciones, el paradigma orientado a objetos.

### 5.7 Paradigma Orientado a Objetos

El paradigma orientado a objetos es una disciplina de ingeniería de desarrollo y modelado de software que permite construir más fácilmente sistemas complejos a partir de componentes individuales. Objetos + Mensajes = Programa.

#### Objetos

Los objetos son entidades que se pueden distinguir de manera clara y definida, ejemplos de un objeto pueden ser una persona, un libro, un botón de orden.

La estructura interna de un objeto está compuesta por tres elementos fundamentales como son:

- **Propiedades:** Son las características observables de un objeto. Las propiedades se reconocen porque describen un aspecto del objeto que se puede medir con una escala establecida previamente. A cada propiedad se le debe asignar un valor el cual permite identificar de manera única al objeto.
- **Métodos:** Se define como un conjunto de acciones que un objeto puede realizar para conseguir un propósito. Los métodos representan la parte viva e interesante de un objeto y se emplean habitualmente para modificar las propiedades del objeto. Al modificar cualquiera de las propiedades del objeto se altera su apariencia y se genera un cambio que el usuario de la aplicación puede percibir.
- **Eventos:** Todos los objetos se relacionan con el mundo que los rodea, esto significa que ningún objeto está aislado y siempre está relacionado con otros objetos. Los eventos son los estímulos que un objeto ejerce sobre otro

**Estado de un objeto.** El estado de un objeto es la apariencia que el objeto presenta al usuario, y depende del valor que tenga sus propiedades. Un cambio de estado se logra alterando al menos una de las propiedades del objeto.

Algunos conceptos fundamentales de la programación orientada a objetos se revisarán a continuación:

La tecnología OO permite utilizar de manera óptima las características de los objetos como son el diseño de componentes, mantenimiento y reutilización, los conceptos que permiten lograr esta orientación son los siguientes:

- a. **Identidad:** Los datos están cuantificados en entidades discretas y distinguibles llamadas objetos.
- b. **Clasificación:** Significa que los objetos con la misma estructura de datos (atributos) y comportamiento (operaciones) se agrupan para formar una clase.
- c. **Polimorfismo:** Significa que una misma operación puede comportarse de modos distintos en distintas clases.

- d. Herencia: Compartir atributos y operaciones entre clases tomando como base una relación jerárquica.

Para comprender un poco a qué se refiere esta forma de visualizar un problema se definen los siguientes conceptos:

**Clases.** Los objetos están organizados en familias claramente delimitadas. Una familia se reconoce porque reúne un grupo de objetos que heredan elementos entre sí. El modelo de programación visual, al igual que la naturaleza, organiza los objetos en clases (familias), así una clase es el conjunto de objetos que pertenecen a una misma familia. Por ejemplo, Madonna y Michael Jackson son objetos de una clase cantantes de rock, sin embargo, personas específicas con nombres específicos son miembros de esa clase si poseen ciertas características comunes.

**Herencia.** Un objeto es heredero de otro cuando posee todas sus propiedades, todos sus métodos y reconoce todos sus eventos, aunque pueda disfrutar de propiedades, métodos y eventos adicionales. Se define la herencia como la característica que tienen los objetos de derivarse unos de otros. Por otra parte, la herencia supone una clase base y una jerarquía de clases que contienen las clases derivadas de la clase base, así las clases derivadas pueden heredar las propiedades y métodos de una clase base, añadiendo sus propios métodos y propiedades, incluso cambiar aquellos elementos de la clase base que necesiten sean diferentes.

Existen dos tipos de herencia

- Herencia Simple: En esta jerarquía cada clase tiene como máximo una sola superclase. La herencia simple permite que una clase herede las propiedades y métodos de su superclase en una cadena jerárquica.
- Herencia múltiple: Una malla o retícula consta de clases, cada una de las cuales pueden tener dos o más superclases inmediatas. Una herencia múltiple es aquella en la que cada clase puede heredar las propiedades y métodos de cualquier número de clases.

**Encapsulamiento.** Un objeto está separado del medio que lo rodea por algún tipo de envoltura. Esta separación es la que determina la unidad del objeto, es decir, lo convierte en algo independiente. La envoltura oculta los detalles relacionados con la construcción interior del objeto, esto significa que sólo se conoce del objeto aquello que es revelado por sus métodos. Se llama encapsulamiento a la propiedad que tienen los objetos de ocultar detalles internos, así se permite asegurar que el contenido de la información de un objeto esté oculta al mundo exterior (por ejemplo un objeto A no conoce lo que hace un objeto B y viceversa). Por otra parte, el programador que construye un objeto tiene acceso a todas sus partes: propiedades, métodos y definición de eventos. El programador que utiliza un objeto sólo tiene acceso a sus propiedades y métodos (si estos son públicos) y a la programación de los eventos que han sido definidos para el objeto.

Algunos métodos y propiedades en un objeto son privados, es decir, útiles únicamente dentro del objeto y desconocidos por quien utiliza el objeto, los restantes métodos y propiedades son públicos, los programadores los emplean para cambiar el estado del objeto.

**Comunicación entre objetos (mensajes).** El mensaje es esencialmente una orden que se envía a un objeto para indicarle que realice alguna acción. También se define mensaje como la información completa que un objeto emisor transmite hacia un objeto receptor.

**Polimorfismo.** El polimorfismo es la propiedad que indica literalmente, la posibilidad de que una entidad tome muchas formas, en términos prácticos el polimorfismo permite referirse a objetos de clases distintas mediante el mismo elemento de programa y realizar la misma operación de diferentes formas, según sea el objeto que se referencia en ese momento. Por ejemplo, cuando se describe la clase mamíferos se puede observar que la operación comer es una operación fundamental en la vida de los mamíferos, de modo que cada tipo de mamíferos debe poder realizar la operación o función de comer. Por otra parte, una vaca o una cabra que pasta en el campo, un niño que se come un bombón o caramelo y un león que devora a otro animal, son diferentes formas que utilizan los distintos mamíferos para realizar la misma función (comer).

Un lenguaje de programación que cumple con todos estos conceptos y que se estudiará un poco más a fondo en el siguiente capítulo es JAVA.

El desarrollo de software ha hecho que los programadores tengan que especializarse y conocer diversas herramientas para tener un amplio panorama y poder elegir cuál herramienta es la adecuada para un determinado proyecto. Y hay un campo enorme de explotación actualmente con las redes de datos y sobre todo Internet, ya que ahora las aplicaciones que se diseñan tienen que compartir recursos y colocar algoritmos de seguridad, etc. En el siguiente capítulo se hablará lo que respecta al desarrollo de aplicaciones en Internet y como el concepto de multimedia es otra opción de ofrecer información de forma interactiva con el usuario final.

## Capítulo 6

### Desarrollo de Aplicaciones en Internet Multimedia y Manejo Digital de Imágenes

Siguiendo con el objetivo que persigue la ingeniería de software que se refiere al desarrollo de software de alta calidad, se estudiarán las tecnologías de desarrollo para contenidos dinámicos e interactivos en la Web.

Internet sin duda revolucionó las comunicaciones y el cómputo en todo el mundo, es por ese motivo que se analizarán cuáles fueron las razones para que este fenómeno se diera y las tecnologías que se desarrollaron en su entorno. Por otro lado, se puede hacer uso de Multimedia para que las aplicaciones no sólo sean funcionales, sino también atractivas e interactivas. Para iniciar el presente capítulo se verán algunos conceptos básicos.

#### 6.1 Conceptos Básicos

La información mostrada en la red inicialmente se basaba en texto, imágenes y vínculos (páginas estáticas) sin tener ninguna interactividad con el usuario, sin embargo, la necesidad fue creciendo y solicitando más recursos hasta hoy, donde se pueden visitar páginas donde se accede a base de datos, se puede participar en encuestas, comprar productos y muchos servicios más.

#### Internet

Internet se puede definir como una red global de redes de computadoras cuyo propósito es permitir el intercambio libre de información entre todos sus usuarios. Pero sería un error pensar en Internet únicamente como una red de computadoras.

Actualmente se debe considerar a teléfonos móviles, cajeros de bancos, electrodomésticos, televisores, teléfonos fijos y casi cualquier cosa ser parte de Internet.

#### Historia de Internet

Los inicios de Internet [29] nos remontan a los años 60's. En plena guerra fría, Estados Unidos creó una red exclusivamente militar, con el objetivo de que en el hipotético caso de un ataque ruso, se pudiera tener acceso a la información militar desde cualquier punto del país. Esta red se creó en 1969 y se llamó ARPANET. Al inicio, la red contaba con 4 computadoras distribuidas entre distintas universidades del país. Dos años más tarde, ya contaba con unas 40 computadoras conectadas. Tanto fue el desarrollo de la red que su sistema de comunicación se quedó obsoleto. Entonces se creó el Protocolo TCP/IP, que se convirtió en el estándar de comunicaciones dentro de las redes informáticas (actualmente se sigue utilizando este protocolo).

ARPANET siguió en crecimiento y abriéndose al mundo, cualquier persona con fines de investigación o académicos podía tener acceso a la red. Las funciones militares se desligaron de ARPANET y fueron a parar a MILNET (red creada por los Estados Unidos). La NSF (National Science Fundation) crea su propia red informática llamada NSFNET, que más tarde absorbe a ARPANET, creando así una gran red. El desarrollo de las redes fue enorme, y de esta manera se crean nuevas redes de libre acceso que más tarde se unen a NSFNET, formando lo que hoy se conoce como INTERNET.

La evolución de la NSFNET fue tal que para el año 1990 ya contaba con alrededor de 100,000 servidores conectados. Algunos de los servicios que se encuentran disponibles en Internet aparte de la WEB, son el acceso remoto a otras máquinas (telnet), transferencia de archivos (FTP), correo electrónico (SMTP), conversaciones en línea (IMSN MESSENGER), transmisión de archivos (P2P), etc.

### ¿Cómo funciona Internet?

Internet es, básicamente, millones de computadoras conectadas entre sí independientes unas de otras. Todas estas computadoras para que puedan coexistir y comunicarse entre ellas, deben tener reglas. Con este motivo fueron creados los Protocolos, que son reglas o normas de comunicación que han de adoptarse para ser entendidas por otras computadoras de la red. Los protocolos más importantes en una red de computadoras son Protocolo de Control de Transmisión (Transfer Control Protocol) y el Protocolo de Internet (Internet Protocol). Usualmente se trata a estos dos protocolos como si fueran uno solo, TCP/IP. Para que una computadora pueda comunicarse en la red debe manejar estos dos protocolos.

Todas las computadoras que componen Internet no están conectadas unas con otras, sólo se están conectando con la más cercana. Si se realiza una petición de cierta información a una computadora que se encuentra a cientos de kilómetros, se accede formando una especie de cadena. Para saber donde está la computadora a la que se quiere acceder, se utilizan las direcciones IP, que son como una especie de número de teléfono único compuesto por una combinación de cuatro números, cada uno formado entre 0 y 255 (por ejemplo, una dirección IP podría ser: 155.210.13.45), ésta se utiliza para identificar a cada computadora dentro de una red.

Una vez sabido esto, la información que se ha solicitado a la computadora lejana, llega al cliente en pequeños paquetes que una vez en la computadora local, se enlaza mostrándose en pantalla.

El servicio más utilizado y aprovechado por Internet son las Páginas Web, las cuales son documentos electrónicos que contienen información específica de un tema en particular y que es almacenado en algún sistema de cómputo que se encuentre conectado a Internet, de tal forma que estos documentos puedan ser consultados por cualesquier persona que se conecte a esta red mundial de comunicaciones y que cuente con los permisos apropiados para hacerlo. Una página Web es la unidad básica del World Wide Web.

Una característica de una página Web es que el texto se combina con imágenes para hacer que el documento sea dinámico y permita que se puedan ejecutar diferentes acciones, una tras otra, a través de la selección de hipervínculos (texto remarcado o de las imágenes), acción que nos puede conducir a otra sección dentro del documento, iniciar un mensaje de correo electrónico, abrir otra página Web o transportarse a otro Sitio Web totalmente distinto.

A diferencia de una Página los *Sitios Web* son un conjunto de archivos electrónicos y páginas que se refieren a un tema en particular, que tienen una página inicial a la que se le llama home page, con un nombre de dominio y dirección en Internet específicos.

Un Sitio Web no necesariamente se localiza en el sistema de cómputo de manera local. Los documentos que forman el Sitio Web pueden encontrarse en un equipo de otra localidad, inclusive de otro país. Lo único que debe cumplirse es que el equipo en el que residan los documentos esté conectado a la red mundial de Internet. Este Servidor Web, como se le denomina técnicamente, puede contener más de un sitio Web y atender concurrentemente a los visitantes de cada uno de los diferentes sitios.

Los Sitios Web requieren de una dirección en particular al igual que los edificios, oficinas y casas, para que los usuarios puedan acceder a la información contenida en ellos. Esta dirección, o URL [30] (Uniform Resource Locator), es un acrónimo de localizador uniforme de recursos, permite localizar o acceder de forma sencilla, cualquier recurso de la red desde el navegador de la WWW.

Otro término muy utilizado en lo que se refiere a Internet es *Portal*, el cual se refiere a un Sitio Web que sirve o pretende servir como un sitio principal de partida para las personas que se conectan al World Wide Web. Son sitios que los usuarios tienden a visitar como sitios de inicio. Los portales tienen gran reconocimiento en Internet por el poder de influencia que tienen sobre grandes comunidades.

El objetivo que se pretende es emplear estos portales para localizar la información y los sitios que interesan y de ahí comenzar alguna actividad en Internet. Un Sitio Web no es un portal sólo por tratarse de un sitio robusto o por contener información relevante. Un portal es una plataforma de inicio para la navegación en el Web.

Para poder consultar información en Internet no es necesario recordar los millones de IP's de host's disponibles en la red, para esto sólo recordamos su Dominio.

Un *Dominio* o nombre de dominio es el nombre que identifica un sitio Web. Cada dominio tiene que ser único en Internet. Por ejemplo, "www.unam.com.mx" es el nombre de dominio de la página Web de la UNAM. Un solo servidor Web puede servir múltiples páginas Web de múltiples dominios, pero un dominio sólo puede apuntar a un servidor.

Un dominio se compone normalmente de cuatro partes: en www.unam.com.mx, las tres uves dobles (www), el nombre de la organización (unam), el tipo de organización (com) y el país al que pertenece (mx). Los tipos de organización más comunes son .COM, .NET, .MIL, y .ORG, que se refieren a comercial, network, militar, y organización (originalmente sin ánimo de lucro, aunque ahora cualquier persona puede registrar un dominio .org).

Puesto que Internet se basa en direcciones IP, y no en nombres de dominio, cada servidor Web requiere de un servidor de nombres de dominio (DNS) para traducir los nombres de los dominios a direcciones IP. Cada dominio tiene un servidor de nombre de dominio primario y otro secundario.

Los dominios se establecen de acuerdo al uso que se le da al sitio Web y al lugar donde se encuentre. Ejemplo:

- .com Comercial
- .edu Educación
- .gob Gobierno
- .mx México

Los dominios a su vez se van dividiendo en otros dominios:

- .gob.mx Gobierno de México
- .com.mx Comercio en México

La asignación de dominios está regulada por organismos nacionales e internacionales para evitar la duplicidad de direcciones; para el caso de México el organismo al que compete la regulación es NIC México, cuya dirección en Internet es [www.nic.mx](http://www.nic.mx).

Existe una entidad sin ánimo de lucro, la ICANN, que coordina todos los dominios del mundo. Digo “coordina”, porque no los gestiona directamente, sino que sólo supervisa cómo otras entidades, como NIC México con los dominios .mx por ejemplo, administran los diferentes tipos de dominios.

Para poder satisfacer la solicitud de peticiones de páginas Web es necesario tener un Servidor Web, el cual es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

El programa de aplicación más difundido es apache (<http://www.apache.org/>), el cual mantiene el 60% de los servidores Web en el mundo.

En cambio para poder realizar solicitudes Web es necesario un Cliente Web o browser, la cual es una aplicación de software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores Web de todo el mundo a través de Internet.

Los navegadores más usados en la actualidad son:

- Internet Explorer de Microsoft
- Navigator de Netscape



Un concepto que no se debe pasar por alto y que es muy importante es el que se refiere a las *Aplicaciones Web*, las cuales los usuarios usan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los webmails, tiendas en línea son ejemplos bien conocidos de aplicaciones Web

Por lo tanto, se pueden definir a las aplicaciones Web como un sistema de información donde una gran cantidad de datos volátiles, altamente estructurados, son consultados, procesados y actualizados mediante navegadores.

Algunos ejemplos de tipos de aplicaciones Web de acuerdo a su función son:

- Informativas (noticias, educativas, etc.)
- Orientadas a descarga de datos (servidores de archivos, música, etc)
- Interactivas (encuestas)
- Orientadas a servicio (simuladores)
- Transaccionales (comercio electrónico)
- Orientadas al análisis de datos (aplicaciones de almacenes de datos, datawarehouse)

Para continuar con el tema enseguida se verán las tecnologías de desarrollo que existen y de las cuales se apoyan los programadores para desarrollar importantes aplicaciones.

### 6.2 Tecnologías de desarrollo para Internet

Ahora se revisarán las tecnologías que se utilizan para Internet y poder, simplemente, consultar una página Web o incluso acceder a bases de datos o programas con nuestro mismo cliente o navegador Web. En seguida se analizarán algunas tecnologías, iniciando, quizá, con la más popular y utilizada: el HTML.

#### HTML

HTML [31] (Hyper Text Markup Language) *Lenguaje de Marcas de HiperTexto*, no es en realidad un lenguaje de programación como Java, Perl, C, o BASIC; es mucho más simple. Es una manera de describir cómo un conjunto de texto e imágenes deben desplegarse en el navegador.

Su creador Tim Berners-Lee fue quien en 1991 presentó a la comunidad científica una manera rápida de poder consultar información y actualizar con facilidad.

El lenguaje está diseñado a través de *etiquetas (tags)* y *atributos* que trabajan conjuntamente para dar alguna característica específica a la página Web; el navegador interpreta dichas etiquetas y atributos, y las despliega en la pantalla.

Para poder realizar páginas Web con HTML sólo es necesario contar con:

- Un editor de texto (bloc de notas, Wordpad en Windows; para Unix VI o Pico, y en el caso de Mac, SimpleText o Teachtext ).
- Un navegador, que puede ser Netscape Navigator o Microsoft Internet Explorer.

El lenguaje no describe la apariencia del diseño de un documento, sino que ofrece a cada plataforma que le de formato según su capacidad y la de su navegador (tamaño de la pantalla, fuentes que tiene instaladas, etc.).

Fundamentalmente, los documentos escritos en HTML están formados por texto propio del documento y las tags que pueden llevar atributos. Esto llevado a la práctica, vendría a ser:

```
<tag> texto afectado </tag>
```

La etiqueta (tag) del principio activa la orden y la última (que será la del principio precedida del signo /) la desactiva. No todas las tags tienen principio y final, pero esto se verá más adelante.

Así, un simple archivo HTML "hola, mundo", con título, sería:

```
<html>

<head>
<title>Hola, mundo</title>
</head>

<body>
Hola, mundo.
</body>

</html>
```

La intención no es explicar todas las etiquetas que utilizan HTML sino más bien dar las características principales de la tecnología.

En seguida se analizará cómo es el proceso que se lleva para poder visualizar una página Web. En primer lugar, cada documento debe tener un localizador o un identificador, algo que diga cómo se llama el documento y dónde se encuentra (URL). En segundo lugar, debe existir un lenguaje en que las computadoras se comuniquen para pedirse y entregarse documentos unas a otras (HTTP HyperText Transfer Protocol, Protocolo para la Transferencia de Hipertexto). Y en tercer lugar, debe existir una forma de codificar los documentos para que una vez que un ordenador, sea el que sea, lo haya obtenido y sea capaz de representarlo en la pantalla o en cualquier otro medio (HTML).

Es así como se puede observar que estos conceptos tienen mucho que ver en las consultas Web que se realizan, no se debe olvidar a TCP/IP que es quien está encima de todos ellos y quien es quien en realidad establece la comunicación con otro equipo en la red y HTTP estaría en capa 7 (Presentación).

Sin duda, HTML hizo de la Web una potente herramienta de consulta, pero no es más que un subconjunto de una tecnología que actualmente se está desarrollando con más fuerza, fortaleciendo el tema de seguridad en las aplicaciones, esta tecnología es XML.

### XML

XML [32] (eXtensible Markup Language) no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

En teoría HTML es un subconjunto de XML especializado en presentación de documentos para la Web, mientras que XML es un subconjunto de SGML especializado en la gestión de información para la Web. En la práctica XML contiene a HTML aunque no en su totalidad.

XML fue creado al amparo del Word Wide Web Consortium (W3C) organismo que vela por el desarrollo de WWW partiendo de las amplias especificaciones de SGML.

Respecto a sus objetivos que persigue son:

- XML debe ser directamente utilizable sobre Internet.
- XML debe soportar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- Los documentos XML deben ser legibles por humanos y razonablemente claros.
- El diseño de XML debe ser preparado rápidamente.
- El diseño de XML debe ser formal y conciso.
- Los documentos XML deben ser fácilmente creables.

Aunque, a primera vista, un documento XML y un documento HTML puedan ser muy parecidos, hay una diferencia principal y es que el HTML no indica lo que se está representando, tiene datos mal definidos con elementos de formato, mientras que el XML contiene datos que se autodefinen: describe el contenido de lo que etiquetan.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, y casi cualquier cosa que se pueda pensar. Ahora como un lenguaje de programación, sin duda Java ha tenido enormes contribuciones en el desarrollo de aplicaciones sobre redes, y ha sido tan importante que se revisará este lenguaje enseguida.

### Java

Sun Microsystems, líder en servidores para Internet, es quien ha inventado el lenguaje Java [33], nace como resultado de resolver simultáneamente todos los problemas que se planteaban a los desarrolladores de software por la proliferación de arquitecturas incompatibles, tanto entre las diferentes máquinas como entre los diversos sistemas operativos y sistemas de ventanas que funcionan sobre una misma máquina, añadiendo la dificultad de crear aplicaciones distribuidas en una red como Internet.

Java es el primer lenguaje que tiene la virtud de ser compilado e interpretado de forma simultánea. Es compilado cuando un programador realiza una aplicación o un applet en Java y lo compila. El compilador Java únicamente genera el denominado *ByteCode*. Este código es un código intermedio entre el lenguaje máquina del procesador y Java. Evidentemente este código no es ejecutable por sí mismo en ninguna plataforma hardware, pues no corresponde con el lenguaje de ninguno de los procesadores que actualmente se conocen. Por lo tanto, para ejecutar una aplicación Java es necesario disponer de un mecanismo que permita ejecutar el ByteCode. Este mecanismo es la denominada *Máquina Virtual Java*. En cada plataforma (Unix, Linux, Windows 95/NT, Macintosh, etc.) existe una máquina virtual específica. Así que cuando el ByteCode llega a la máquina virtual, ésta lo interpreta pasándolo a código máquina del procesador donde se esté trabajando, y ejecutando las instrucciones en lenguaje máquina que se deriven de la aplicación Java. De este modo, cuando el mismo ByteCode llega a diferentes plataformas, éste se ejecutará de forma correcta, pues en cada una de esas plataformas existirá la máquina virtual adecuada. Con este mecanismo se consigue la famosa *multiplataforma* de Java, que con sólo codificar una vez, se puede ejecutar en varias plataformas.

Sin embargo, existen algunas desventajas, y la más clara es la velocidad de ejecución. Puesto que la máquina virtual debe estar interpretando constantemente el ByteCode, se consume demasiado tiempo de procesador en realizar esta interpretación, que por otra parte no aporta nada a la aplicación, obteniendo así un bajo rendimiento. Para solucionarlo se han adoptado soluciones intermedias. Una de las más novedosas y útiles son los compiladores *JIT* (Just-In-Time). Estos compiladores están situados a la entrada de la máquina virtual, de forma que según llega el ByteCode lo van compilando al lenguaje máquina del procesador. A diferencia de la interpretación, el compilador no ejecuta el ByteCode, únicamente lo traduce y lo almacena en código nativo dentro de la máquina virtual. Así, una vez que la aplicación está dentro de la máquina virtual, ya se encuentra en lenguaje máquina y, por lo tanto, será directamente ejecutable, sin necesidad de interpretaciones, consiguiendo dotar de mayores rendimientos a la aplicación.

Las características principales que nos ofrece Java respecto a cualquier otro lenguaje de programación, serían:

- **Lenguaje simple**

Java posee una curva de aprendizaje muy rápida. Resulta relativamente sencillo escribir applets interesantes desde el principio. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros.

- **Orientado a objetos**

Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos.

- **Distribuido**

Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

- **Interpretado y compilado a la vez**

Java para conseguir ser un lenguaje independiente del sistema operativo y del procesador que incorpore la máquina utilizada, es tanto interpretado como compilado. El código fuente escrito con cualquier editor se compila generando el ByteCode. Este código intermedio es de muy bajo nivel, pero sin alcanzar las instrucciones máquina propias de cada plataforma. Ese mismo código es el que se puede ejecutar sobre cualquier plataforma. Para ello hace falta el runtime (sistema de ejecución en tiempo real), que sí es completamente dependiente de la máquina y del sistema operativo que interpreta dinámicamente el ByteCode. Con este sistema es fácil crear aplicaciones multiplataforma, pero para ejecutarlas es necesario que exista el runtime correspondiente al sistema operativo utilizado.

- **Robusto**

Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

- **Seguro**

Dada la naturaleza distribuida de Java, los applets se bajan desde cualquier punto de la Red, la seguridad se impuso como una necesidad de vital importancia. A nadie le gustaría ejecutar en su ordenador programas con acceso total a su sistema, procedentes de fuentes desconocidas. Así que se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.

- **Indiferente a la arquitectura**

El código fuente Java se "compila" a un código de bytes de bajo nivel independiente de la máquina. Este código (ByteCode) está diseñado para ejecutarse en una máquina hipotética que es implementada por un sistema run-time, que sí es dependiente de la máquina.

- **Portable**

Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, Pc o Mac.

- **Multihebra**

Hoy en día ya se ven como terriblemente limitadas las aplicaciones que sólo pueden ejecutar una acción a la vez. Java soporta sincronización de múltiples hilos de ejecución (*multithreading*) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

- **Dinámico**

El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la red.

A través del tiempo se han desarrollado potentes entornos de desarrollo de software por los que había que pagar costosas licencias como Visual Studio de Microsoft. A partir de hace un tiempo se puede contar con una herramienta con potencia similar y de libre distribución. El nombre de esta herramienta es *Eclipse*, una plataforma de herramientas universal con un Entorno Integrado de Desarrollo (IDE) abierto y extensible, para cualquier cosa y nada en particular.

### **Eclipse**

A pesar de que Eclipse [34] está escrito en su mayor parte en Java (salvo el núcleo), se ejecuta sobre máquina virtual de ésta y su uso más popular se da como un IDE para Java, Eclipse es neutral y adaptable a cualquier tipo de lenguaje de programación, por ejemplo C/C++, Cobol, C#, XML, etc.

Eclipse tiene una característica clave y es la extensibilidad. Eclipse es una gran estructura formada por un núcleo y muchos plug-ins (un plug-in es la mínima unidad de la plataforma que puede ser desarrollado por separado y que la aporta una nueva funcionalidad.) que van conformando la funcionalidad final. La forma en que los plug-ins interactúan es mediante interfaces o puntos de extensión; así, las nuevas aportaciones se integran sin conflictos ni dificultad.

El Proyecto Eclipse es un proyecto de desarrollo de software de código abierto dedicado a proporcionar una plataforma industrial robusta, con amplias características y con calidad comercial para el desarrollo de herramientas altamente integradas.

Por lo tanto Eclipse es una plataforma que se adapta perfectamente para programar en Java.

Tal como se ha revisado Java es un lenguaje de programación muy potente y en pleno desarrollo. Mediante la creación de *applets* en Java se puede descargar un programa desde el servidor hasta el cliente y ejecutarse de manera local y bajo una página HTML.

Un *applet* es un pequeño programa hecho en Java que es descargado automáticamente como parte de una página Web, al igual que cualquier gráfico, y cuando se activa, ejecuta un programa. Esto hace especialmente atractivo un applet, ya que se puede distribuir software al cliente desde el servidor, en el momento en que el cliente necesite ese software, y no antes, con lo cual siempre tendrá el cliente la última versión de ese software, se actualice cuando se actualice. Además, tal como está diseñado Java, el programador necesita crear su programa una sola vez, y ya estará listo para ser ejecutado en todas las plataformas que dispongan de un navegador con soporte Java.

Para poder llamar un applet en una página Web solo se utiliza la etiqueta <APPLET>. Un ejemplo sencillo para llamar a un applet desarrollado en Java llamado "miapplet.class" sería de la siguiente manera:

```
<HTML>
<APPLET CODE="miapplet.class" HEIGHT=100 WIDTH=300></APPLET>
</HTML>
```

Una tecnología que Java incorpora son los Servlets que nos permiten crear páginas Web de manera dinámica en base a peticiones. Un Servlet es un programa creado en Java que a diferencia de un applet, este se ejecuta en el servidor Web. Un servlet es un programa que recibe peticiones y genera a partir de ellas páginas Web de manera dinámica.

### Otras tecnologías de programación

En Internet cada día se demandan nuevas formas de comunicarse y de compartir recursos es por esa razón que existen diferentes tecnologías que se desarrollaron y otras que se están desarrollando, algunas otras toman características de otras y convergen para tratar de fortalecerse y estandarizarse como es el caso de Java.

Uno de los primeras formas de programación Web dinámica fueron los CGI's (Common Gateway Interface). Su función principal es la de añadir una mayor interacción a los documentos Web que por medio del HTML que se presentan de forma estática. Estos programas son desarrollados en PERL principalmente y tienen la característica que son interpretados y como desventaja actual es que se corren en el servidor Web provocando una carga de trabajo gradual conforme haya clientes solicitando peticiones.

Netscape por su parte desarrollo un lenguaje llamado JavaScript con el fin de incrementar las funcionalidades de una página Web. Algunas características importantes de JavaScript son las siguientes:

- Es un lenguaje interpretado y no compilado. El Navegador del usuario se encarga de interpretar el código y ejecutarlo.
- Es un lenguaje orientado a eventos.
- Es un lenguaje orientado a objetos.

JavaScript no es propiamente un lenguaje de programación como lo es Java y de hecho su nombre solo se le colocó para llamar el interés de los programadores pero en realidad son muy diferentes.

Algunas desventajas de utilizar JavaScript:

- JavaScript no puede crear gráficos:
- JavaScript no pueden crear sus propios gráficos, solamente puede manipular gráficos hechos en otros programas.
- JavaScript trabaja de forma diferente en diferentes navegadores. Este es el principal problema de JavaScript, su comportamiento depende del navegador.

Un lenguaje que también se desarrollo interpretado y no compilado fue Visual Basic Script pero con la gran desventaja que solo es compatible en Internet Explorer.

VBScript o Visual Basic Scripting es similar a los lenguajes Visual Basic que utilizan los desarrolladores de Windows y tiene, esencialmente, la misma estructura que JavaScript.

Visual Basic Script se ejecuta del lado del cliente pero también se puede desarrollar programas que se ejecuten del lado del servidor Web de Microsoft (IIS). Este es el caso de las páginas ASP, en donde el código lo interpreta y ejecuta el servidor Internet Information Server antes de enviarlo al cliente que solicitó la página.

Tal como se ha revisado Java actualmente es un lenguaje de programación de gran potencial y en pleno desarrollo.

Ahora existen herramientas que ayudan a integrar los servidores Web y los manejadores de bases de datos que ya se revisaron en el anterior capítulo, lo cual traería como resultado una aplicación integral y sumamente poderosa.

### Análisis de Integración entre servidores Web y manejadores de base de datos

La integración de las tecnologías existentes está dirigida a satisfacer las necesidades y demandas de los usuarios. Inicialmente era suficiente transmitir información mediante páginas HTML, solo que únicamente era estático y para este fin funcionaron perfectamente los Servidores Web (Web Servers) como Apache. El usuario solicitaba una página Web mediante HTTP al Web Server y este a su vez le enviaba de igual forma mediante HTTP un archivo estático que era interpretado por el navegador del cliente y mostrado al usuario. Esto lo podemos ejemplificar con la figura 51.



Figura 51.- Solicitud de página Web tradicional

Sin embargo con el tiempo fue necesario implementar algún mecanismo que pudiera enviar algo más que un archivo estático al cliente. Lo que se hizo fue que en el mismo Servidor de páginas se llamara a un intérprete (generalmente PERL), como lo muestra la figura 52, que ejecutaba un “script” (programa) y enviara los resultados nuevamente al Servidor de Páginas y este lo enviara al cliente. Sin embargo la desventaja es que los procesos se ejecutaban en mismo Host del Servidor de Páginas y si las solicitudes eran mínimas no había problemas en cambio si había solicitudes masivas el servidor tendía a caer.

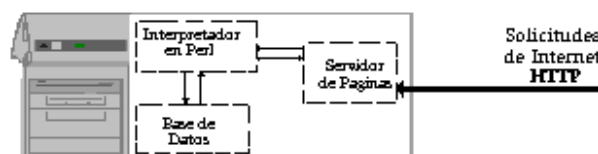


Figura 52.- El servidor llama a un intérprete (PERL)

Java proporciona una solución para mejorar el uso de los recursos de los Servidores de Páginas, hace uso de los Contenedores (Servlets Engines).

Un Servlet es utilizado para extender las capacidades de un servidor de páginas Web y así generar páginas HTML dinámicamente y para poderse ejecutar es necesario tener un contenedor de servlets el cual trabaje junto con el servidor de páginas.



El contenedor administrará la carga y descarga de servlets. Tomcat proporciona el contenedor necesario para el trabajo con Servlets como lo muestra la figura 53.

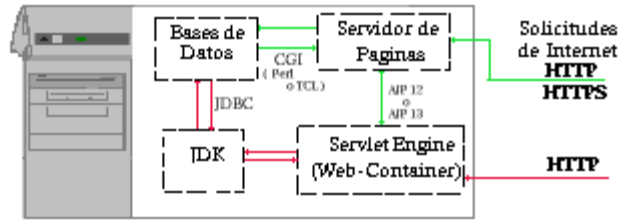


Figura 53.- Uso de Contenedores o Servlets

En la figura anterior la línea roja demuestra que el servlet engine (también llamado "Web Container") es capaz de responder a requisiciones de Internet, en efecto actuando como "Servidor de Páginas", sin embargo, aunque esto sea posible la gran mayoría de las implementaciones de Servlet Engines no funcionan tan eficiente como un "Servidor de Páginas", es por esto que se opta por utilizar un "Servidor de Páginas" (Apache,Aol,Netscape) en conjunción con un "Servlet Engine".

La línea verde demuestra la utilización de un "Servidor de Páginas" con un "Servlet Engine" (Web-Container), esta implementación suele utilizarse cuando se requiere utilizar encriptación o se tiene un sitio que contiene documentos puros en HTML , las cuales son dos áreas que un "Servidor de Páginas" supera en desempeño comparado con un "Servlet Engine".

**¿Que hace el Servlet Engine?.** El "Servlet Engine" ofrece un "Ambiente" donde habitan los JSP y Servlets, es ahí donde se contemplan una gran cantidad de funcionalidades como: threading, administración de sesiones, conectividad con el "Servidor de Páginas", es por esto al "Servlet Engine" también se le denomina "Web-Container".

Ahora falta integrar los EIS (Sistemas de Información Empresarial) y es aquí donde los *Servidores Aplicativos* (Application Servers) cobran gran importancia y así integrar las tecnologías vistas. Estos "Application Servers" comunmente llamados Middleware se encuentran compuestos como lo muestra la figura 54:

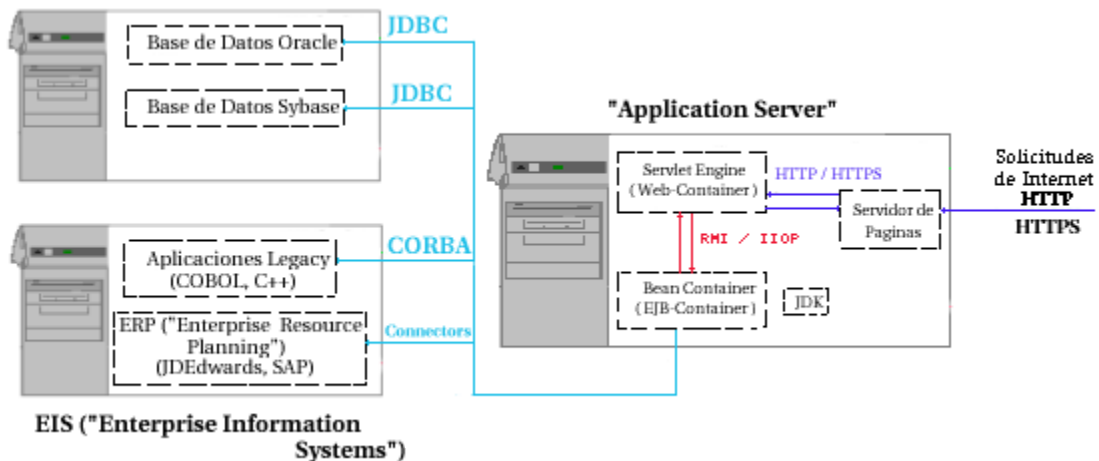


Figura 54.- Uso de Application Server

Como su denominación lo implica ("Middleware") se encuentran en la parte media de una arquitectura de sistema, su flexibilidad y gran importancia reside en la posibilidad de acceder a la información en sistemas empresariales (EIS) como SAP, JdEdwards, bases de datos o alguna aplicación escrita en COBOL u otro lenguaje. Los dos elementos primordiales son el "Servlet Engine" (Web-Container) y "Enterprise Bean Engine" (Bean-Container).

- El Servlet Engine (Web-Container) en un "Application Server" realiza las mismas funcionalidades que fueron mencionadas anteriormente.(Ofrecer un ambiente para JSP y Servlets).
- El "Enterprise Bean Engine" (Bean-Container) ofrece un "ambiente" donde residen EJB's ("Enterprise Java Beans") , es mediante "Enterprise Java Beans" que se ejecuta la lógica de negocios sobre la información que reside en los sistemas empresariales ("EIS").

Algunos "Application Servers" son:

- WebLogic
- Websphere
- Oracle 9i Application Server
- Sun Application Server

Al final no se debe dejar de tomar poca importancia al tener la necesidad de generar un sistema, tal como se ha revisado se deben considerar bastantes cosas, es decir se tiene que formar una arquitectura acorde al diseño ya planeado.

### 6.3 Arquitecturas

En el tema de arquitecturas es sumamente importante que al momento de diseñar un sistema, re revise lo siguiente:

- Cual es la mejor arquitectura para esta aplicación
- Cual es la mejor herramienta para desarrollar lo que pide el cliente
- Como se diseñará la base de datos

Con lo anterior se puede generalizar que todas las aplicaciones se dividen en tres partes, las cuales se deben revisar detenidamente para su posterior diseño:

- Interfaz de usuario
- Reglas de negocio
- La forma de acceder a los datos

Para poder revisar estas tres partes existen tres tipos de arquitecturas que se muestran a continuación:

**Arquitectura mono-capa.** Son aquellas que las tres partes (Interfaz de usuario, reglas de negocio y acceso a datos) se encuentran en la misma máquina y se administran por la misma herramienta. Lo podemos ejemplificar en la figura 55.

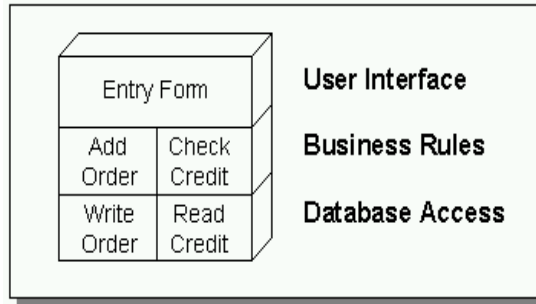


Figura 55.- Arquitectura Mono-capa

**Arquitectura en dos capas (Two-Tier).** Son aplicaciones conocidas como Cliente/Servidor (Figura 56).

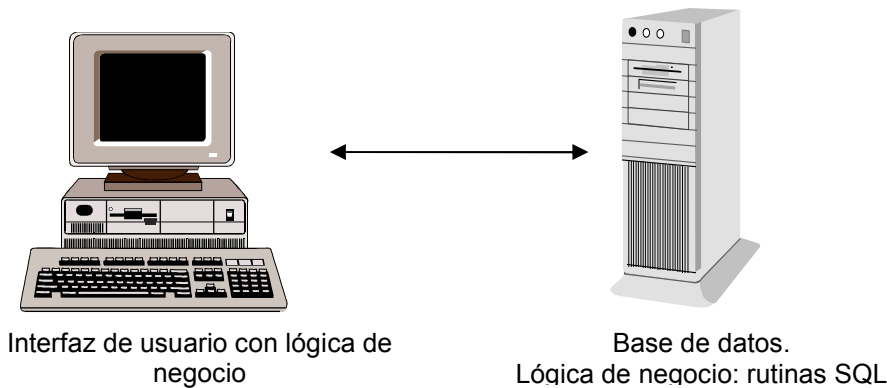


Figura 56.- Arquitectura en dos capas (Two-Tier)

**Arquitectura en tres capas (Three-Tier).** Cada elemento de la aplicación se separa en una sola entidad (Figura 57).

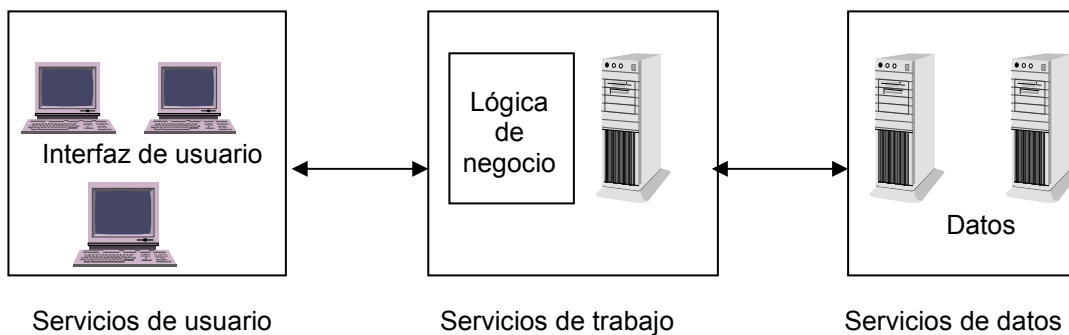


Figura 57.- Arquitectura en tres capas (Three-Tier)

Hasta este momento se han revisado algunas tecnologías que se aplican para poder acceder a diferentes recursos en la red. Ahora se estudiará un poco sobre Multimedia y como este concepto fue ligándose desde un inicio para poder tener presentaciones enriquecidas no solo de texto como se hacia a principios de los 80's con el MS-DOS, sino ahora con sonidos, videos, ambientes agradables y sobre todo la aportación que tuvo en Internet y los Sistemas Operativos modernos. En el siguiente punto se analizará lo que se refiere a esto.

## 6.4 Conceptos Básicos de Multimedia

Multimedia [35] quiere decir “múltiples medios” y el concepto es profundo y antiguo ya que tiene relación con la comunicación y la cultura. Es decir para poder transmitir conocimiento o como medio de comunicación como se describió en el capítulo 1 se dieron varias formas desde las pinturas rupestres encontradas en cuevas hasta hoy día el uso de la PC e Internet.

En lo que se refiere específicamente a multimedia computacional se tendrá que remontar a la aparición de la computadora en plena segunda guerra mundial cuando en 1943 los británicos desarrollaron una computadora llamada COLUSSOS desarrollada para fines bélicos y se tiene que revisar como poco a poco se fue desarrollando la tecnología del hardware y software para que Apple introdujera por primera vez la interfaz gráfica en el mundo de la computación, una manera diferente de ver el computo para esa época.

Y con la aparición de Internet se dio un paso gigantesco para el desarrollo de aplicaciones multimedia aplicando tecnologías como Java para poder comunicarse y compartir toda clase de recursos. Existen diferentes tipos de sistemas y podemos hacer una clasificación.

## 6.5 Tipos de Sistemas

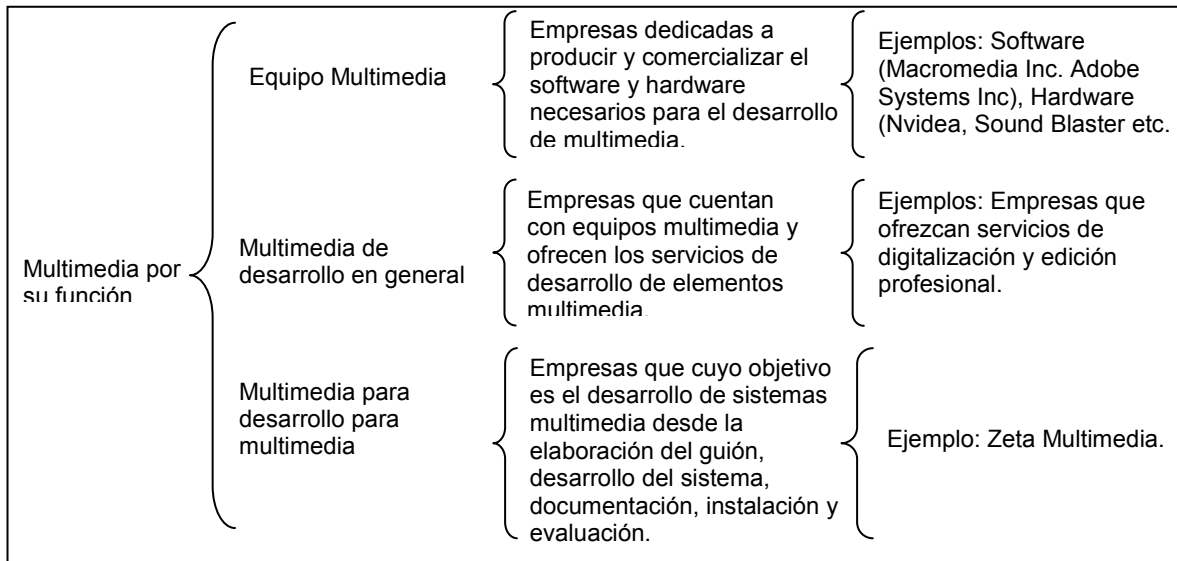
Existen diferentes tipos de sistemas multimedia, en términos generales los podemos dividir en base al uso que desempeñan:

- **Funcionales.** Están orientados a servicio público permitiendo a los usuarios consultarlos disipando dudas o simplemente funcionan como informativos. Por ejemplo en algunas agencias de automóviles existen terminales con sistemas interactivos junto a los autos que nos permiten consultar características como tipo de motor, potencia, equipamiento, precio, etc.
- **Didácticos.** Este tipo de sistemas se enfocan a fines educativos o de capacitación profesional. Por ejemplo podemos encontrar sistemas que ayudan a las personas con capacidades diferentes a crecer profesionalmente o simplemente como apoyo a terapias de rehabilitación.
- **Persuasivos.** Estos se enfocan a la propaganda y publicidad de productos y servicios valiéndose para ello de todo tipo de medios como sonidos, imágenes sensuales con impactos psicológicos.
- **Entretenimiento.** Este tipo de sistemas tienen que ver con juegos sin tener necesariamente que ser educativos, en el mercado podemos encontrar miles de ellos y existen empresas dedicadas exclusivamente a este campo ya que es muy rentable.

Una manera más de clasificar los tipos de sistemas es en base a la función que desempeñan.

## 6.6 Multimedia por su Función

Los sistemas multimedia se pueden clasificar en base a su función que realizan, y para expresar esta información se concreta en el cuadro 7:



Cuadro 7.- Multimedia por su Función

En el desarrollo de sistemas multimedia existen diferentes elementos que constituyen toda la aplicación, algunos de ellos se verán enseguida.

## 6.7 Elementos Multimedia

Existen diferentes tipos de elementos que se utilizan para poder transmitir información o simplemente realizar una comunicación entre dos puntos, algunos de ellos son los siguientes:

### Texto

El texto sin duda es un elemento básico en la comunicación o transmisión de algún mensaje o información y no solo se refiere a el texto en claro sino hay que estudiar si el medio será impreso o a través de un monitor, color de la fuente tamaño, posición, etc.

### Imagen Fija

Las imágenes enriquecen el documento o la información que se desea transmitir. Actualmente existen diferentes maneras de obtener imágenes, ya sea creándolas con algún software, digitalizándolas o incluso ya podemos bajarlas desde las cámaras digitales o teléfonos móviles que ya cuentan con cámaras integradas.

Las imágenes pueden ser de dos tipos: mapas de bits (complejas, detalles finos) o dibujos basados en vectores (Figuras matemáticas).

### Imágenes en Movimiento

Las imágenes en movimiento o también llamados videos e incluso animaciones, no son más que imágenes mostradas en sucesión de una manera rápida para de alguna manera engañar al cerebro humano y percibir videos de la vida cotidiana. Los videos actualmente toman gran importancia ya que para fines educativos y comerciales son perfectos.

Existen tres tipos de animaciones por computadora: la cinematográfica, la animación virtual por computadora y la creada por programación.

## Audio

El sonido enriquece a las imágenes en movimiento transmitiendo juntos mucha más información. Sin embargo por si solo el sonido es sumamente importante ya que el simplemente escuchar música o la radio nos hace transportarnos a los lugares en el pensamiento. Existen diferentes herramientas para convertir el sonido análogo a sonido digital utilizando muestreos y acercándose lo más posible al sonido real aunque solo sea un acercamiento.

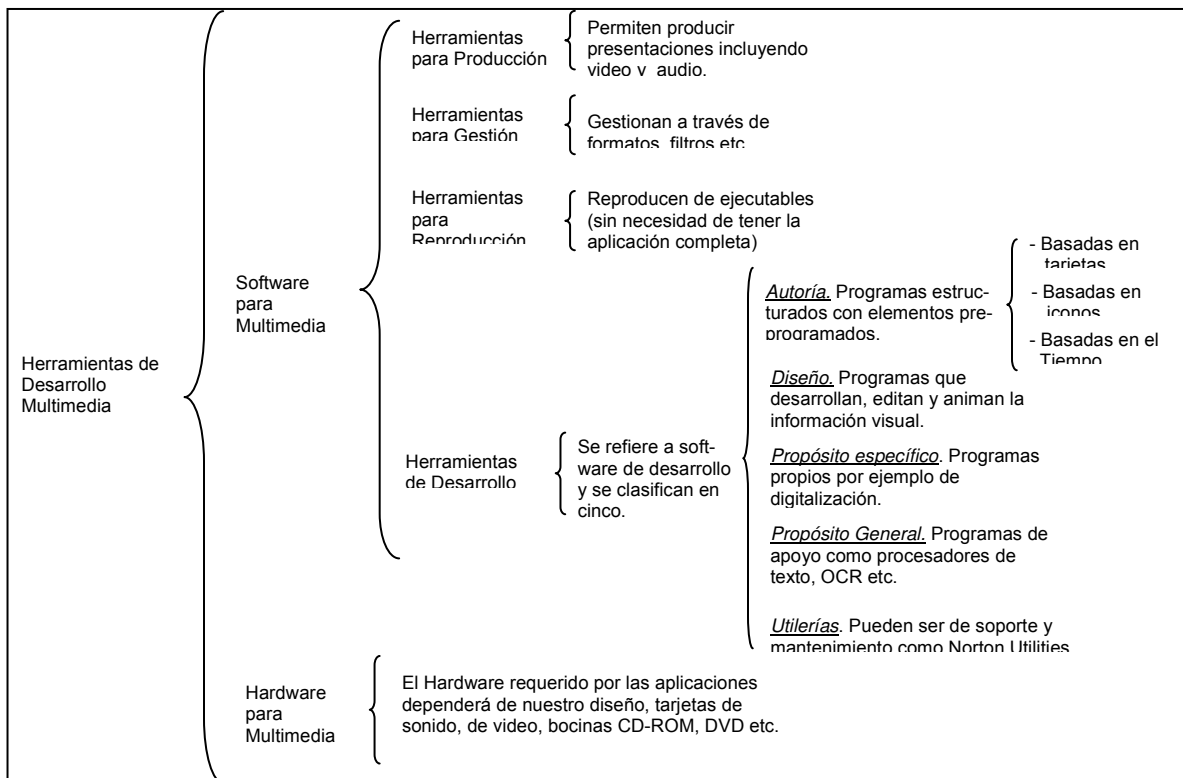
## Hipertexto

Hipertexto es un término que se utiliza para referirse a algo más que solo texto claro. Un hipertexto es un documento digital que se puede leer de manera no secuencial. Es decir se puede elegir el orden en que se quiere leer la información. Un hipertexto tiene los siguientes elementos: secciones, enlaces y anclajes. Las secciones o nodos son los componentes del hipertexto. Los enlaces son las uniones entre nodos que facilitan la lectura secuencial o no secuencial del documento. Los anclajes son los puntos de unión entre nodos.

Enseguida se revisarán que tipo de herramientas existen para su creación.

## 6.9 Herramientas de Desarrollo

Existe una gran variedad de herramientas para el diseño de proyectos multimedia, para saber que herramienta utilizar dependerá de que tipo de proyecto se necesite, enseguida a través del cuadro 8, se pueden ver los diferentes tipos:



Cuadro 8.- Herramientas de desarrollo Multimedia

Con todo lo anterior se tiene un amplio panorama de herramientas de desarrollo, que da un amplio margen a un líder de proyecto de elegir las mejores aplicaciones y herramientas amoldándose a las necesidades de los clientes. Ahora se verán las necesidades actuales de mercado. Actualmente como se revisó se desarrollan aplicaciones basadas en redes ó concretamente Internet, la principal demanda es para realizar comercio electrónico por la red permitiendo bajar los costos a las empresas por no tener corporativos enormes físicamente hablando, o para bajar costos por no tener la necesidad de almacenar mercancías a grandes escalas. Todo ello se analizará en el siguiente capítulo.

## Capítulo 7

### Comercio Electrónico y Administración de Proyectos de TI

Las diferentes tecnologías de información y el crecimiento de Internet han propiciado que el comercio se lleve a cabo de manera diferente a lo tradicional, actualmente para ser una empresa comercializadora ya no es necesario tener grandes instalaciones físicas o tener grandes capitales, solo basta con hacer uso de las tecnologías y la red global llamada Internet. Es así como surge el concepto llamado comercio electrónico. En el presente capítulo se verá como se realiza esta práctica, de que herramientas se apoya el comercio electrónico para que aporte seguridad a las transacciones, y como se relaciona con otras tecnologías.

En la segunda parte del capítulo se estudiara como la administración de proyectos aplica los conocimientos, habilidades, herramientas y técnicas sobre las actividades de un proyecto para lograr el cumplimiento de los requerimientos del producto o servicio, es decir como podemos administrar el desarrollo de un proyecto aplicando todos los conocimientos vistos en los anteriores capítulos.

#### 7.1 Introducción al Comercio Electrónico

El comercio electrónico [36] genera grandes ventajas sobre el comercio tradicional, por ejemplo: las barreras geográficas ya no son problema, comunicación rápida y barata, costos sumamente bajos, capacidad de compra y venta alta, capacidad de penetración alta. Por otro lado se agudizan algunos problemas y surgen otros más, como por ejemplo la seguridad en las transacciones, contratos sin papel, protección a los derechos de propiedad intelectual entre otros.

A través del tiempo han surgido diferentes formas de hacer comercio. A principios de los 20's en Estados Unidos surge la venta por catálogo utilizada por las grandes empresas de mayoreo, consistía en acercar un catálogo de fotos de productos a personas en zonas rurales, en este catálogo se colocan características, fotos, precios de los productos ofertados; los clientes tienen la ventaja de poder escoger en la comodidad de su casa sin presiones de vendedores. Esto permitió extender el alcance del mercado a zonas lejanas sin necesidad de abrir sucursales o tiendas que generan gastos.

Para los 80's la televisión adopta el mismo concepto, solo que ahora el mercado o clientes potenciales son más lejanos geográficamente. El método ahora es mostrar por medio de la TV las características de los productos y la forma de cobro ahora es mediante una llamada telefónica usando las tarjetas de crédito.

Con la aparición en 1989 de la www se expandió el mismo método solo que ahora los clientes potenciales están por todo el mundo realizando operaciones de compra y venta en línea mediante las tecnologías de información existentes.

Comercio electrónico se refiere a cualquier forma de transacción comercial (no sólo comprar y vender) en la que las partes interactúan electrónicamente en lugar de tener presencia física.



El comercio electrónico se apoya en diferentes tecnologías, algunas son las siguientes:

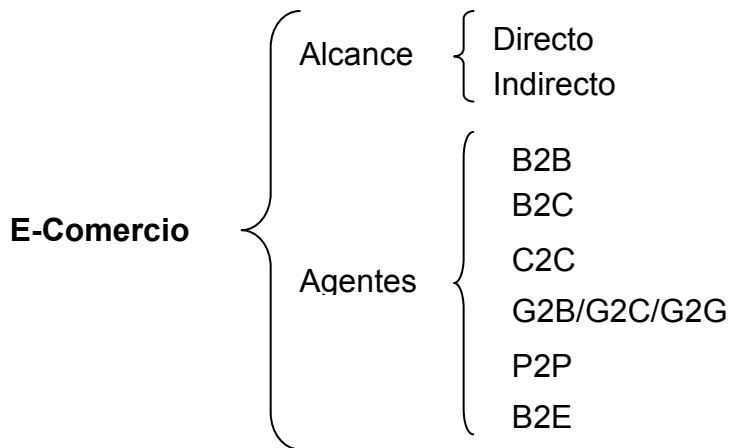
- Intercambio electrónico de datos (EDI)
- Correo electrónico (e-mail)
- Transferencia electrónica de fondos (EFT)
- Aplicaciones de voz: buzones, servidores, telefonía IP
- Transferencia de archivos (FTP)
- Diseño y fabricación por computadora (CAD/CAM)
- Multimedia
- Tablero electrónicos de publicidad
- Videoconferencia

### Clasificación del comercio electrónico

Se puede clasificar el comercio electrónico de acuerdo a diferentes perspectivas, una de ellas es de acuerdo a su alcance y se divide en 2: Directo e Indirecto.

El comercio electrónico indirecto se refiere a cuando se realiza la selección y pedido del producto por medios electrónicos y se envían al cliente por los medios tradicionales, en cambio el comercio electrónico directo tanto el proceso de selección como el de envío se realizan de manera electrónica, para que esto ocurra las mercancías deben ser intangibles como música, libros, revistas, etc.

Otra forma de clasificar al comercio electrónico es de acuerdo a los agentes que intervienen como lo muestra el cuadro 9:



---

Cuadro 9.- Tipos de comercio electrónico

- **De empresa a empresa (B2B).** Comercio directo de empresa a empresa en una relación cliente / servidor.
- **De empresa a consumidor (B2C).** Venta directa de empresa a consumidor.
- **De consumidor a consumidor (C2C).** Usuario final se pone en contacto para vender con otro usuario final.
- **De gobierno a empresa / ciudadano / gobierno (G2B / G2C / G2G).** Transacciones entre empresas, ciudadanos y gobierno.
- **Punto a punto (P2P).** Sin fines de lucro a diferencia de C2C.

- **De empresa a empleado (B2E).** Posibilita la comunicación entre la empresa y el empleado.

Existen algunos factores que motivaron el desarrollo de las empresas para poder comercializar electrónicamente, por ejemplo: el adelantarse a sus competidores utilizando las tecnologías actuales, el saber que los competidores ya utilizan el comercio electrónico, o el echo de que los proveedores, empresas o usuarios finales ya utilicen la tecnología, sin embargo este tipo de comercio genera ventajas y desventajas:

Ventajas:

- Canal de publicidad y comunicación disponible las 24 horas teniendo un alcance masivo.
- Se reducen intermediarios ya que se puede tratar directamente con el usuario final.
- Se reducen costos al sustituir procesos o mercancías físicas por digitales.

Desventajas:

- Se tiene que invertir en tecnología nueva y capacitación.
- La seguridad y privacidad siguen siendo un ladrillo pesado que frenan el crecimiento exponencial de la tecnología.
- Legalidad política y sociedad. Existen problemas en la regulación internacional de esta manera de comerciar.

Un tema que desde luego es muy importante, “La Seguridad”, y digo que es importante porque tal como se revisó en el capítulo 3, existe un canal sucio en la red donde existen miles de usuarios escuchando, existen algunas herramientas que se desarrollan para tratar de eficientar el problema, uno de ellos es EDI.

### 7.2 EDI

El Intercambio Electrónico de Datos (EDI) es la comunicación de forma estructurada de computadora a computadora y de aplicación a aplicación (sin intervención humana y sin papel) entre corporaciones, instituciones o individuos, utilizando un formato estándar para los documentos de negocios entre las empresas.

Usuarios de EDI intercambian información y transacciones con sistemas computacionales propios, en lugar de hacerlo por correo electrónico o por algún medio público o general.

Para poder generar una transacción vía EDI las empresas deben considerar lo siguiente:

- Mínimo hardware necesario (PC)
- Interfaz de comunicación que dependerá de la forma en que tengamos el enlace (MODEM, red, inalámbrico, etc).
- Un programa (software) de traducción EDI que permitirá que la información fluya entre los sistemas como un lenguaje común de acuerdo al estándar.
- Un programa de comunicaciones, el cual se encarga de establecer la comunicación con la Red de Valor Agregado (VAN).

Un problema que se genera al comerciar electrónicamente es generar facturas, ¿Cómo se realiza esto?, enseguida se mostrará que hay soluciones.

### Factura Electrónica

A fin de poder generar una factura electrónica con la misma validez oficial que la tradicional se formó un comité en noviembre de 1997 integrado a la estructura del comité mexicano de comercio electrónico (COMECE) teniendo participación empresas de telecomunicaciones (Avantel), comerciales e industriales (Gigante, Palmolive, Kellogg's, etc.) y proveedores de tecnología (Infosel, Levicom, Seguridata etc.).

De igual forma tienen participación miembros del comité financiero, de seguridad y el grupo impulsor de legislación para el comercio electrónico (GILCE) y la participación técnica de organizaciones internacionales, dirección de proyectos jurídicos e internet.

El comité de factura electrónica desarrolló el proyecto en varias etapas, algunas concluidas y otras se encuentran en proceso.

- Información que debe contener la factura electrónica, se analizó una factura tradicional para definir que campos mínimos debe contener de manera electrónica.
- Uso de estándar internacional de Intercambio Electrónico de Datos (EDIFACT – EANCOM versión D 96A) estándar que proporciona seguridad al mensaje.
- Desarrollo de guías de implantación del mensaje de la factura electrónica (INVOIC).
- Cambios necesarios en la legislación. A partir de las reformas publicadas el 29 de marzo del 2000 se reconoce que los mensajes de datos tienen la misma validez legal que los documentos escritos.
- Se han realizado pruebas piloto para validar que todos los elementos involucrados en este proyecto cumplan con su objetivo.
- Se ha impulsado el reconocimiento de la factura electrónica como un documento legal y físicamente válido.
- Publicación de la norma de la conservación de datos electrónicos.
- Ley de firmas electrónicas.

Y como tema indispensable es la Seguridad que debe existir, es por ello que se estudiarán algunas formas que ayudan a minimizar riesgos.

### Seguridad

El tema de seguridad en el comercio electrónico es de suma importancia ya que las transacciones realizadas deben ser seguras.

Tal como se revisó en el capítulo 3 de Seguridad Informática, al tener que viajar la información por un canal promiscuo el riesgo es muy alto. Sin embargo actualmente la criptografía a desarrollado algoritmos complejos y junto con la firma electrónica y los certificados de calidad nos permiten tener una transacción relativamente “Segura”.

Por ejemplo si se quiere cuidar la Autenticación y la Integridad de la información al enviar un mensaje de manera relativamente segura sería de la siguiente forma: Al texto en claro se le aplica una función hash (para no cifrar el mensaje completo ahorrando tiempo y carga de trabajo al procesador), el resultado se cifra utilizando la llave pública del destinatario que

previamente fue alojada y validada por una autoridad certificadora (Certificado Digital), avalando que la llave publica pertenece al destinatario, El resultado será la Firma Digital que será enviada junto con el mensaje original en claro.

Al llegar al destinatario el mensaje y la firma, aplica la función Hash al mensaje en claro que llevo y descifra la firma digital con su llave privada, deben coincidir los resultados de la función hash para que sea seguro que el mensaje no fue alterado y que al poderlo descifrar con la llave privada se pueda decir que lo envió la persona que dice ser.

Es necesario recordar que dependerá de que cosa se quiera cifrar, si se quiere cuidar la autenticación, integridad, confidencialidad, control de acceso o no repudio. El tema de seguridad no se ha implementado como se quisiera, ya que como lo hemos estudiado existen barreras y costos que estancan esta labor tan importante. Ahora una aplicación creada realiza peticiones sin que intervenga el usuario y para eso tenemos los Servicios Web, los cuales actúan sin que aparentemente el usuario lo sepa.

### 7.3 Servicios Web

Una manera de entender lo que es un servicio Web es hacer una analogía con lo que es una función o rutina en un programa; el programa realiza una solicitud a una función esperando un resultado o acción y no importa como se creo o que proceso realice.

Un servicio Web permite compartir información o ejecutar alguna tarea por solicitud de diferentes aplicaciones, no importando el sistema operativo, plataforma, o dispositivos que se utilicen. Un servicio Web viene a ser una función o rutina en Internet y a diferencia de una página Web, es solicitado únicamente por aplicaciones y no por usuarios.

Existen diferentes servicios Web ya creados y listos para utilizarse principalmente por programadores Web ya que están ligados a Internet. Los protocolos que soportan los Servicios Web (Webservices) se comunican normalmente por el puerto 80 basándose en http (métodos Get y Put).

Existen dos protocolos utilizados por los Servicios Web basados en XML los cuales son: XML-RPC y SOAP (Simple Object Access Protocol) y antes de diseñar un Servicio Web hay que decidir cual utilizar ya que no son compatibles entre ellos. ¿En base a que hay que decidir?, bueno la respuesta esta ligada a que lenguaje de programación invocara al Servicio Web, por ejemplo .NET de Microsoft trabaja con SOAP.

Existen diferencia entre ambos protocolos, sin embargo la principal es la complejidad. XML-RPC es más fácil de utilizar y un programador novato puede sacar buenos resultados sin ser experto en el tema, sin embargo esto trae como consecuencia que XML-RPC no sea tan potente como SOAP, el cual requiere mayor complejidad pero el programador podrá controlar más variables.

Bueno existen empresas que ya operan con algunos procesos que si bien no son los más actuales funcionan, pero tienen la necesidad de actualizarse y migrar todas las tecnologías a las actuales.

## 7.4 Tecnologías comunes en comercio electrónico

Desde luego, el comercio electrónico ha generado un alto impacto en las empresas, pero para poder acceder a esta tecnología es necesario transformar completamente las reglas del negocio. Para llevar a cabo esta transformación se han desarrollado diferentes tecnologías, de las cuales pueden echar mano las empresas, algunas de ellas son: CRM, SCM, KM, BI y ERP.

- **CRM.** (Customer Relation Management, Administración de la Relación con el Cliente). Es una metodología que permite identificar, adquirir y retener clientes facilitando la administración y coordinación de los clientes a través de canales múltiples, departamentos, líneas de negocio y geografía.
- **SCM** (Suplly Chain Management, Administración de la Cadena de Suministros). Los Sistemas de la Cadena de Suministros permiten administrar las operaciones de compra críticas de una organización. El SCM permite sustituir el tedioso inventario tradicional donde se tenía que tener grandes volúmenes de mercancía acumulada por desconocer el movimiento del mercado; con el SCM se trabaja de diferente forma utilizando lo que se conoce como B2C (Business to Consumer) donde se acercan los proveedores con el cliente final (CRM).
- **KM** (Knowledge Management, Administración del Conocimiento). La KM consiste en transformar el conocimiento táctico (el que sabe un trabajador específico) en explícito (conocimiento documentado y replicable) y así convertirlo en un activo estratégico de la organización.

La figura 58 muestra cómo pueden hacerse negocios electrónicos involucrando CRM, SCM y KM.

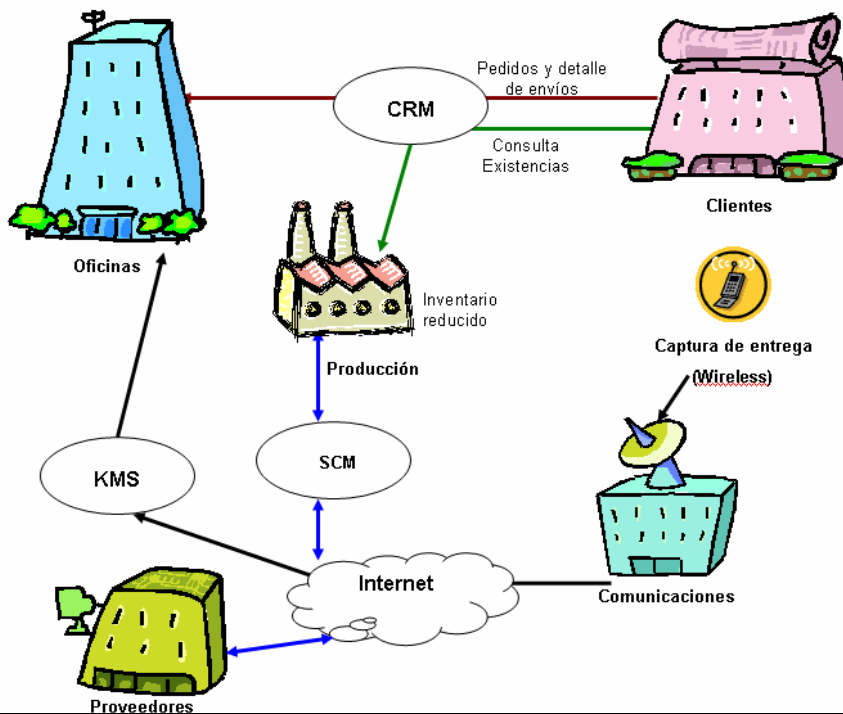


Figura 58.- Esquema de Negocios Electrónicos

En el esquema muestra como el CRM se encarga de interaccionar con el cliente, levantando pedidos, haciendo un seguimiento de las operaciones, hacer envíos y consultar existencias entre otras cosas, por otra parte el SCM, a través de Internet, conecta al área de producción con los proveedores satisfaciendo las necesidades de insumos para la producción, al final los sistemas KM retroalimentan todo el sistema aportando todo el conocimiento para poder evaluar el comportamiento del mercado.

- **BI** (Business intelligence, Inteligencia de Negocios). La inteligencia de negocios es un concepto que implica análisis y procesamiento de la información que posee la organización con el fin de obtener ventajas competitivas sobre sus competidores. Una buena estrategia de inteligencia de negocios debe considerar tecnologías como:
  - Data Warehousing y Data Marts
  - Aplicaciones Analíticas
  - Herramientas OLAP
  - Herramientas de consulta y reportes de datos
  - Herramientas de producción de reportes personalizados
  - Herramientas ETL
  - Herramientas de Administración de Sistemas
  - Portales de Información Empresarial
  - Sistemas de Bases de Datos.
  - Sistemas de Administración de Conocimiento
- **ERP** (Enterprise Resource Planning, Planeación de los Recursos Empresariales). El sistema de Planeación de los Recursos Empresariales es una aplicación de gestión empresarial que integra los flujos de información de distintas áreas y tiene la finalidad de dar soporte y colaboración para que se dé una eficiente administración.

Poco a poco se va armando el conocimiento para poder generar aplicaciones, en seguida se revisará qué se requiere.

### 7.5 Plan de Negocio

Una vez revisados los conceptos referentes a comercio electrónico, ya se puede implementar una estrategia. Y para poder lograr esto se debe considerar algunos pasos.

Lo primero que se debe realizar es reservar un nombre de dominio, el cual identificará en la red y para este fin existen muchas empresas entre las cuales se pueden encontrar Network Solutions, Mexico.com, NIC, etc.

Un segundo punto es considerar un sistema, prácticas o aplicación de administración de proyectos, ya que esto ayudará a tener un mejor control de todo el proceso de planeación e implantación, identificando aplicaciones que serán necesarias para la identificación, adquisición y publicidad de nuestro producto.

Las decisiones acertadas a cerca de cómo estará configurada la empresa se pueden obtener a partir de tres procesos: identificación de requerimientos, alineación y planeación.

**Identificación de requerimientos.** Este proceso se refiere a la recolección y análisis de requerimientos y para esto existe un sistema llamado FURPS+, el cual realiza una clasificación de requerimientos.

FURPS+ es un acrónimo que representa los siguientes elementos:

- **Functionality** (Funcional). Define características del producto final y son particulares de cada desarrollo.
- **Usability** (Utilizable). Características estéticas y de consistencia en la interfaz.
- **Reliability** (Confiabilidad). Características como disponibilidad, capacidad de recuperación de fallos.
- **Performance** (Rendimiento). Tiempo de respuesta, tiempo de recuperación, tiempo de inicialización, de cierre etc.
- **Supportability** (Soportable). Características como adaptabilidad, mantenimiento, compatibilidad, escalabilidad, etc.

El símbolo + representa requerimientos de:

- **Diseño**. Especifica las reglas con las que debe desarrollarse.
- **Implementación**. Indicará la forma en cómo se construirá o codificará.
- **Interfaz**. Especifica entidades externas, así como restricciones y formatos de interacción.
- **Físicos**. Características asociadas al hardware.

**Alineación.** El proceso de alineación define las directrices para la evaluación de las alternativas de solución que satisfagan los requerimientos identificados.

Además de la estrategia organizacional y los objetivos a corto plazo, al alinear se debe tener en cuenta elementos como la misión, visión y valores de la empresa que son el reflejo de la imagen que se pretende brindar al cliente con productos y atención. En la figura 59 muestra los elementos que se deben considerar para una correcta alineación.



Figura 59.- Criterios de Alineación de las TI al Negocio

**Planeación.** Este proceso nos permite optar por una solución rápida que permita satisfacer las necesidades inmediatas, o bien, una planeación estratégica que permita conocer un mapa completo y así guiar a corto, mediano y largo plazo. En la figura 60, se muestra un plan de negocio del proyecto completo conociendo el ciclo de ventas y mercadotecnia con un enfoque del cliente.



---

Figura 60.- Ciclo de ventas y mercadotecnia.

En este ciclo podemos distinguir los siguientes elementos:

1. **Cliente.** Se debe identificar al mercado objetivo y sus necesidades.
2. **Publicidad.** Promoción y difusión para atraer al mercado una primera vez, así como los medios para estimular su regreso.
3. **Mercadeo.** Consiste en la identificación de los productos que se ofrecerán así como en la forma en que se posicionarán y se visualizarán por los clientes.
4. **Servicio de Ventas.** Define la manera en que se atenderán las preguntas de los clientes y se resolverán sus posibles problemas.
5. **Transacción de ventas.** Se refiere a la manera en que se incentivará al público para estimular sus compras.
6. **Procesamiento de transacciones.** Los mecanismos para el manejo de órdenes, impuestos, embarques y pago.
7. **Entrega.** La manera y los criterios para turnar las órdenes al área de entregas y distribución.
8. **Servicio Post – Venta.** Define la manera en que se atenderán las preguntas de los clientes y se resolverán los posibles problemas, pero posterior a la realización de la venta.
9. **Mercadotecnia.** Consiste en analizar la información acerca de hábitos y patrones de consumo, así como las tendencias que éste observa y su respuesta a motivadores como promociones y estacionalidad.

El desarrollo del comercio electrónico ha traído grandes beneficios, tanto para las empresas como para los usuarios, ahora el objetivo del presente estudio es poder agrupar todos los conocimientos del presente diplomado y poder tener una visión de todos los conocimientos que son necesarios para realizar aplicaciones de alta calidad.



### 7.6 Contexto de la administración de proyectos

La administración de proyectos es aplicar conocimientos, habilidades, herramientas y técnicas sobre las actividades de un proyecto para lograr el cumplimiento de los requerimientos del producto o servicio. Por otro lado, es importante mencionar que las operaciones de una empresa suelen ser de dos tipos: procesos y proyectos como lo muestra la tabla 15.

Proyectos	Procesos
Realizados por personas	
Restringidos por recursos limitados	
Planeados	
Ejecutados	
Controlados	
Únicos	Repetitivos
Temporales	Continuos

Tabla 15.- Operaciones de una Empresa

En la anterior tabla se muestran los elementos en común y las diferencias de ambas operaciones. Podemos decir que un proyecto es un producto o servicio único y es temporal, mientras que un proceso es repetitivo y continuo.

#### Fases de un proyecto y su ciclo de vida

Los proyectos pueden llegar a ser muy complejos e implicar cierta incertidumbre en su desarrollo, es por esto que para un mejor control se divide en fases. Con esto se logra que al final de cada fase exista un entregable, el cual consiste en un producto tangible y verificable y que pueda aportar conocimiento para decidir al final de cada fase si se puede pasar a la siguiente o hacer correcciones. Es importante conocer el ciclo de vida de un proyecto que permite conocer el inicio y fin de cada fase y detectar si fuera necesario elementos no considerados al inicio, y así poder incorporarlos. La figura 61, muestra las 5 fases del ciclo de vida de un proyecto de propuestas por el *PMI* (Project Management Institute), se puede observar el nivel de actividad de cada fase durante todo el ciclo de vida.

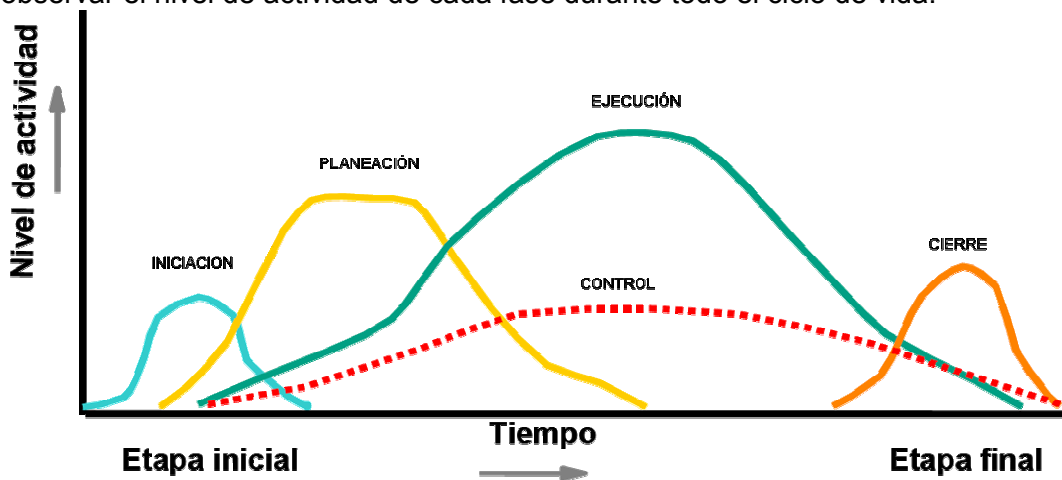


Figura 61.- Fases del ciclo de vida de un proyecto

El conocer la vida de los proyectos puede definir aspectos importantes como por ejemplo:

- Qué trabajo técnico se debe realizar en cada fase.
- Quién debe involucrarse en cada fase (equipos temporales).

En el capítulo 5 se revisó el diseño de sistemas y específicamente en el punto 5.3 se estudió cómo *RUP* (Rational Unified Process) divide el ciclo de vida de los proyectos de TI en 4 fases y 9 flujos de trabajo. En la figura 62, se puede observar las fases propuestas por RUP para diseño de sistemas de TI y se puede observar como el penúltimo flujo de trabajo se refiere a administración de proyectos y está presente durante todo el ciclo de vida teniendo mayor intensidad al inicio y fin.

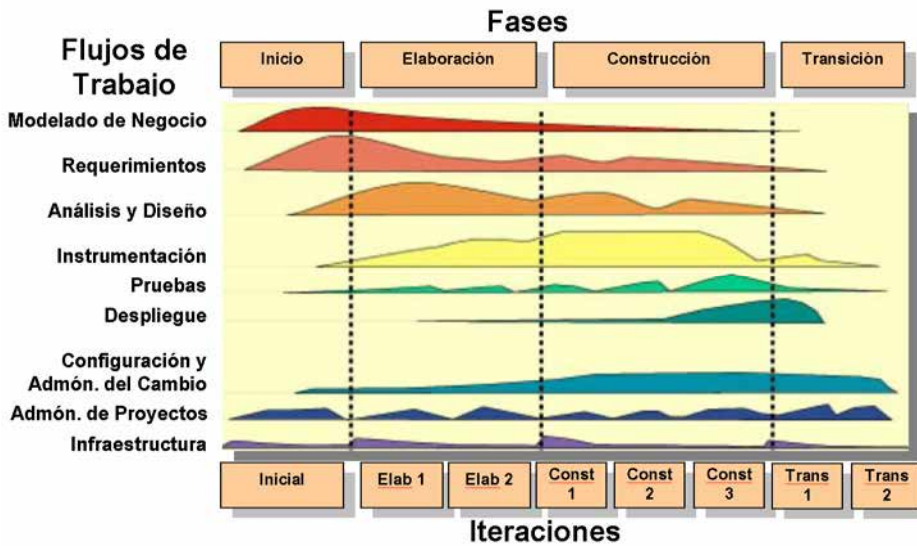


Figura 62.- Ciclo de vida de los proyectos de TI

La correspondencia entre las fases del ciclo de vida de los proyectos de TI, propuestas por RUP y las fases de la administración de proyectos propuestas por el PMI se muestran en la figura 63.

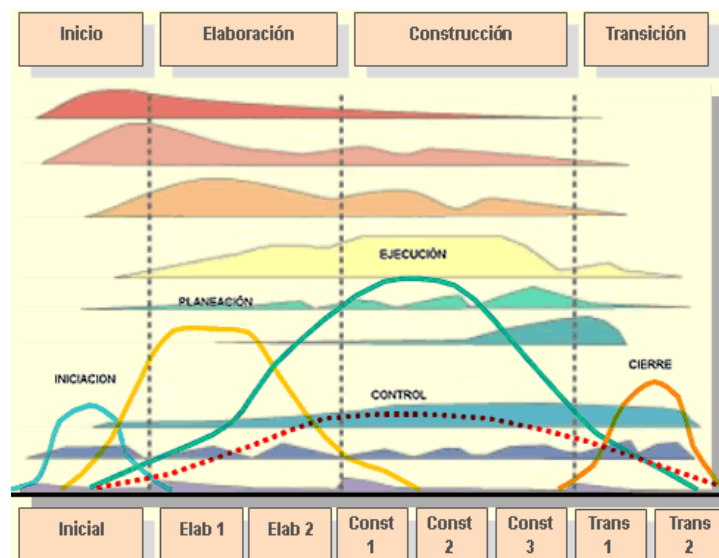


Figura 63.- Correspondencia de ciclos de vida de proyectos RUP-PM

De esta correspondencia se puede observar que ambos coinciden que en la fase de ejecución es donde hay mayor esfuerzo y en el flujo de trabajo referente a la administración de proyectos está presente durante todo el ciclo de vida

Durante el desarrollo del proyecto existen personas y organizaciones que están involucrados y que cuyos intereses están presentes, positiva o negativamente ejerciendo influencias sobre el mismo proyecto, a estos individuos y/o organizaciones se les conoce como Stakeholders (involucrados), algunos de ellos son:

- Administrador del proyecto
- Cliente o destinatario
- Empresa directamente ligada a la realización del proyecto
- Miembros del equipo del proyecto
- Patrocinador, individuo o grupo que provee recursos financieros

### Perfil del Administrador de Proyectos de TI

Se refiere a las habilidades que debe poseer un individuo para ser un buen administrador de proyectos. Algunas de estas habilidades son las siguientes:

- **Liderazgo.** El arte o proceso de influir sobre las personas para que de manera voluntaria y entusiasta se logren las metas del grupo.
- **Comunicación.** Es lograr el correcto intercambio de información entre un emisor y un receptor interviniendo características como tono de voz, volumen, utilizar un lenguaje de acuerdo a las personas con quien se quiera comunicar, etc.
- **Negociación.** Es lograr acuerdos consiguiendo llegar a términos aceptables y llegando al cumplimiento de las metas establecidas.
- **Resolución de problemas.** Involucra trabajo bajo presión, análisis y toma de decisiones.
- **Influencia sobre la organización.** Requiere entender las estructuras formales e informales de todas las organizaciones involucradas. Es la habilidad de poder influenciar el comportamiento, cambiar el curso de eventos y hacer que las personas hagan cosas que de otra manera no harían.

A continuación se revisará cada una de ellas.

### 7.7 Fases de la administración de proyectos de TI

Ahora se revisará el ciclo de vida de un proyecto dividido en 5 fases, el cual es propuesto por el PMI.

1. Iniciación
2. Planeación
3. Ejecución
4. Control
5. Cierre

#### Iniciación

La iniciación es el proceso mediante el que se reconoce formalmente que un nuevo proyecto existe o mediante el cual se avala la continuación de un proyecto existente.

El proceso de iniciación se basará en el estudio de sus entradas, las herramientas y técnicas en que se apoya y las salidas que genera, como se muestra en la figura 64.

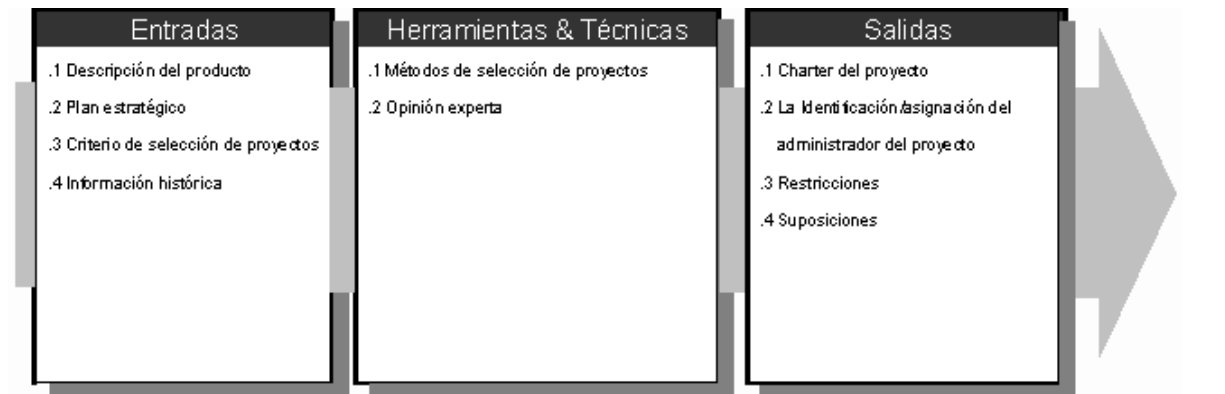


Figura 64.- Proceso de Iniciación

El surgimiento de un proyecto nace típicamente por una necesidad, por ejemplo:

- Una demanda del mercado (Niveles de oferta y demanda).
- Una necesidad del negocio (Identificación de oportunidades, debilidades, fortalezas y amenazas).
- Una demanda de un cliente
- Un avance tecnológico (Incorporar tecnología nueva)
- Un requerimiento legal (Cambios en el marco legal de la empresa)

Se debe considerar que las ideas iniciales deben contemplar cuatro elementos básicos:

1. **Descripción del producto.** Se refiere a especificar las características del producto o servicio que fue elegido para crearse. Una correcta descripción del proyecto debe estar documentada explicando entre otras cosas la relación del producto con la necesidad inicial, al final dicha descripción será soporte para la planeación del proyecto.
2. **Plan estratégico.** El proyecto siempre estará encaminado a cumplir con las metas estratégicas de la empresa (estar alineado), se debe tomar en cuenta puntos importantes como la misión, visión, valores y objetivos generales de la empresa.
3. **Criterios de selección de proyecto.** Son definidos, normalmente, en términos del producto del proyecto y pueden cubrir una amplia gama de categorías administrativas de evaluación.
4. **Información histórica** de decisiones previas de selección de proyectos y de sus reportes de ejecución que deben considerarse siempre que esta información esté disponible.

En esta primera fase es muy satisfactorio echar mano de FURPS+.

### Identificación de requerimientos con FURPS+

Tal como se revisó en el punto 7.5 del presente capítulo, FURPS+ es un sistema de clasificación de requerimientos y en esta primera fase ayudará precisamente a identificar y clasificar los requerimientos del proyecto.

*Un mecanismo arquitectónico* es una solución común a un problema o requerimiento frecuente definido en distintos niveles de abstracción, que permite llegar a la especificación concreta de productos que satisfagan la necesidad identificada.

Por ejemplo, en la tabla 16, con un nivel de especificación de tercer nivel se pueden identificar productos concretos. En el nivel de análisis se definen características genéricas (independientes de la implantación) que debe cumplir el producto, en el nivel de diseño se identifican las tecnologías que cumplen con el requerimiento analítico y al final, en el nivel de instrumentación se especifican las marcas de los productos concretos que cumplen con las necesidades planteadas.

Nivel Análisis	Nivel Diseño	Nivel Instrumentación
Persistencia	RDBMS	Oracle
		Sybase
	OODBMS	ObjectStore
Comunicación	Object request broker	Orbix
		VisiBroker
	Message Queue	MSMQ
		MQSeries

Tabla 16.- Mecanismo Arquitectónico de tres niveles

Para lograr la obtención de requerimientos arquitectónicos se deben seguir algunos puntos básicos:

1. Construir y mantener una lista completa de requerimientos arquitectónicos.
2. Para cada requerimiento de la lista, definir una serie de preguntas que puedan incluirse en el cuestionario para el usuario.
3. En el momento de la entrevista con los usuarios, hacer notar el impacto que pueda tener cada una de las respuestas, para evitar respuestas a la ligera que compliquen el sistema más adelante.
4. Asegurarse que los usuarios asignen una prioridad o peso a cada uno de los requerimientos, para posibilitar la negociación de requerimientos contra los recursos disponibles.

A continuación se mostrarán algunos de los mecanismos de análisis más comunes en el proceso de levantamiento de requerimientos, así como un cuestionario propuesto para cada elemento **FURPS+**.

<b>Functionality (Funcionalidad)</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Auditoria	Provee pista de auditoria, producto de la ejecución del sistema	¿El sistema debe ser auditable?	Complejidad de mantenimiento
Comunicación	Permite la comunicación entre procesos distribuidos.	¿Existirá comunicación entre procesos?	Grado de sofisticación, tiempo de desarrollo, complejidad de mantenimiento.
Depuración	Proporciona elementos que permiten depurar la aplicación.	¿Es necesario incorporar mecanismos para depurar la aplicación?	Tiempo de desarrollo, complejidad de diseño.
Administración de Archivos	Provee servicios para acceso a sistemas de archivos (adjuntar, exportar, importar, etc.).	¿El sistema interactúa con archivos?	Complejidad de la aplicación, tiempo de desarrollo.
Intercambio de Información	Soporte para conversiones de formato en los datos que maneja.	¿El sistema deberá compartir información entre sus módulos?	Complejidad de almacenamiento.

<b>Usability (Utilización)</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Accesibilidad	Facilidad con el cual pueden accederse las funcionalidades del sistema	¿Existe alguna condición especial relativa a la facilidad de uso del sistema (incluyendo usuarios inexpertos)?	Grado de simplicidad, tiempo de desarrollo, complejidad de navegación.
Estética	Calidad y estilo del diseño de la interfaz de usuario.	¿Existe alguna característica especial de apariencia del sistema?	Intervención de especialistas en diseño, tiempo y costo.
Consistencia	Respecto a la forma en que se implementan ciertas funcionalidades en distintos puntos del sistema o en otros sistemas relacionados (F1 siempre para ayuda, por ejemplo).	¿Existe alguna regla de consistencia que deba seguirse?	Grado y nivel de consistencia, tiempo de desarrollo.

<b>Reliability (Confiabilidad)</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Exactitud	Exactitud y precisión con la que se deben efectuar los cálculos.	¿Existen reglas particulares sobre la exactitud o precisión de algún cálculo?	Tipos de cálculos, complejidad de componentes.
Disponibilidad	Porcentaje del tiempo en que el sistema debe estar disponible. "arriba".	¿Cuál es el porcentaje de disponibilidad requerido? En términos de tiempo entre fallas	Redundancia de componentes, incremento de costos.
Recuperación	Capacidad con que el sistema puede recuperarse de fallos.	¿Existe algún requerimiento especial en cuanto a la recuperación del sistema?	Mecanismos sofisticados, complejidad en el desarrollo.

<b>Performance (Rendimiento)</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Tiempo de recuperación.	Tiempo en que el sistema se recupera estabilidad, tras una falla.	¿Existe alguna restricción acerca del tiempo que el sistema debe emplear para recuperarse de una falla?	Reducción del tiempo de recuperación, mecanismos sofisticados.
Tiempo de respuesta.	Lapso en que el sistema responde a las peticiones.	¿Existe alguna especificación sobre el tiempo máximo en que se debe responder al usuario o a eventos particulares?	Optimización de consultas, mecanismos sofisticados.
Tiempo de arranque	Tiempo en que el sistema tarda en inicializarse.	¿Hay alguna definición sobre el tiempo de arranque?	Grado de sofisticación de flujo de datos.

<b>Supportability (Soporte)</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Compatibilidad	Capacidad de compartir información con versiones previas del sistema o con otros sistemas.	¿Es necesario mantener compatibilidad con versiones anteriores o con sistemas legados?	Adaptación de sistemas, complejidad de integración.
Mantenimiento	Facilidad con que pueda darse mantenimiento al sistema.	¿Quién será responsable del mantenimiento?	Grado de amabilidad de la interfaz, tiempo de desarrollo.
Escalable	Facilidad con que el sistema pueda escalarse en términos de usuarios y volumen de datos.	¿Cuál es el plan de crecimiento de usuarios del sistema?	Robustez del sistema, plan de adquisición de recursos.
Diagnósticos	Facilidad con que el sistema puede auto diagnosticarse.	¿Es necesario habilitar esquemas de diagnóstico?	Grado de complejidad de desarrollo.

+

<b>Diseño</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Auditoria	Forma de previsión de pistas de auditoria, producto de la ejecución del sistema	¿El mecanismo de auditoria debe cumplir alguna regla?	Grado de satisfacción del mecanismo.
Comunicación	Forma en que se permite la comunicación entre procesos distribuidos.	¿El mecanismo de comunicación debe cumplir alguna regla?	Grado de satisfacción del mecanismo.
Administración de errores	Permite manejar errores de ejecución, aislarlos y reportarlos.	¿El mecanismo de manejo de errores debe cumplir alguna regla?	Grado de satisfacción del mecanismo, tiempo de desarrollo.
Seguridad	Forma en que se permite generar y aplicar reglas de acceso discrecional a la información.	¿Los mecanismos de autenticación y encriptación deben cumplir alguna regla?	Grado de satisfacción del mecanismo

<b>Instrumentación</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Componentes de terceros	Reglas sobre el uso de componentes de terceros.	¿Existen reglas, incluso de costo, acerca del uso de componentes de terceros?	Búsqueda de componentes recientes, populares de uso común con soporte y mantenimiento.
Lenguajes	Definición de los lenguajes con los que se llevará a cabo la instrumentación.	¿Existe algún lenguaje de programación estándar en la empresa?	Los estándares deben conciliarse con las aplicaciones, se incrementa el tiempo de adquisición y evaluación.
Plataforma	Plataformas que el sistema deberá soportar.	¿Qué plataforma deben soportarse?, ¿Qué sistemas operativos están en uso?, ¿Necesita acceso Web?, ¿Qué tipo de browser?	Un sistema de solo una plataforma simplifica el desarrollo o adquisición.
Límite de recursos	Límites del sistema en cuanto a memoria, disco duro o ancho de banda.	¿Qué límites de memoria, disco, etc. existen?	Incremento en el tiempo de adquisición, complejidad al desarrollo.

<b>Interfaz</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Sistemas expertos	Sistemas expertos con los que el sistema deberá entablar relación.	¿Existen sistemas con los que se requiera establecer interfaz, tanto provista como requerida?	Incremento en tiempo de desarrollo, control de documentación, sistemas de generación de interfaces.
Formatos de Interfaz	Formato en que se debe intercambiar información.	¿Existen definiciones de interacción tales como formatos o protocolos?	El uso de estándares predefinidos simplifica el desarrollo.

<b>Físico</b>			
<b>Mecanismo</b>	<b>Descripción</b>	<b>Pregunta</b>	<b>Impacto</b>
Formato	Formatos del hardware requerido para el funcionamiento del sistema.	¿Existen restricciones de formato del hardware requerido?	Restricciones de formato de dispositivos, especificaciones físicas, hardware propietario, reducción de posibilidades en el mercado.
Tamaño	Tamaño del hardware requerido.	¿Hay alguna restricción del tamaño del hardware?	Restricciones de equipo, mayor tiempo de adquisición.
Peso	Peso del hardware requerido.	¿Hay algún límite de peso que se deba cumplir?	Restricciones de equipo, mayor tiempo de adquisición.



Cuando se recaba toda la información es importante hacer un análisis de requerimientos para poder tomar decisiones respecto al diseño del proyecto. La labor de la toma de decisiones es sumamente importante y suele ser una tarea compleja en base a la enorme cantidad obtenida con FURPS+. Existe una herramienta con la que se puede cuantificar y cualificar la información. Se trata de agregar pesos a cada uno de los elementos de las tablas de FURPS+, dichos pesos son asignados junto con expertos de cada área. En la figura 65, se muestra una matriz de análisis de requerimientos FURPS+.

		Peso (1 -10)	Importancia
F	Auditoría		
	Comunicación		
	Depuración		
	Administración de archivos		
	Gráficos		
	Intercambio de Información		
	Administración de licencias		
	Multilingüístico		
	Correo electrónico		
	Volumen		
	Administración de memoria		
	Meta-datos		
	Ayuda en línea		
	Persistencia		
	Impresión		
	Administración de procesos		
	Reporteo		
	Administración de recursos		
	Calendarización		
	Seguridad		
	Administración del sistema		
Tiempo			
Administración de transacciones			
Flujo de trabajo			
U	Accesibilidad		
	Estética		
	Consistencia		
R	Exactitud		
	Disponibilidad		
	Recuperación		
P	Tiempo de Recuperación		
	Tiempo de Respuesta		
	Tiempo de Baja		
	Tiempo de arranque		
	Throughput		
S	Adaptabilidad		
	Auditable		
	Compatibilidad		
	Configurable		
	Instalable		
	Mantenimiento		
	Escalable		
Diagnósticos			
Totales			

Figura 65.- Matriz de análisis de requerimientos FURPS+

Una vez evaluado y aceptado la aprobación del proyecto se elabora una carta constitutiva con la cual se formaliza la autorización del proyecto. Algunos elementos que debe contener dicha carta son los siguientes, como lo muestra la figura 66:

Administración de Proyectos			
Carta Constitutiva del Proyecto			
CartaCEstacionamientos	Iniciación	Fecha:	
Información del Proyecto			
Fecha:	Descripción:		
Gerente Programa:	Adm. Proyecto:		
Destinatario del Proyecto			
Area Usuaría:			
Patrocinador del Proyecto:	Propietario Proyecto:		
Áreas involucradas:			
Alcance del Proyecto			
Fecha de inicio proyecto:	Fecha de término esperado:	Recursos aprobados:	Costo total estimado:
Descripción:			
Otros proyectos relativos:			
Aceptación Área Propietaria Solución:		Aceptación:	
Fecha aceptación:		Fecha aceptación:	

**Figura 66.- Carta Constitutiva del Proyecto**

1. Identificación del proyecto
2. Identificación del equipo de administración de proyectos.
3. Identificación de área usuaria, patrocinador y propietario del proyecto.
4. Alcance del proyecto, supuestos y restricciones.
5. Proyectos relativos.
6. Fechas de inicio y finalización del proyecto.
7. Costo total del proyecto y recursos aprobables.
8. Firmas de autorización del proyecto.

Después de tener la mayor cantidad de información se pasará a la siguiente fase: Planeación.

### 7.8 Planeación

En la fase de planeación se realiza una selección de objetivos y acciones para lograrlos, es decir, es muy importante en esta fase la toma de decisiones, ya que de aquí depende el éxito del proyecto.

En esta fase se desarrolla el mayor trabajo, ya que aunque ya se hayan desarrollado proyectos similares, siempre es único y hay que arrancar desde cero.

En la figura 67 muestra los procesos que existen en esta fase, los cuales se revisarán más adelante.

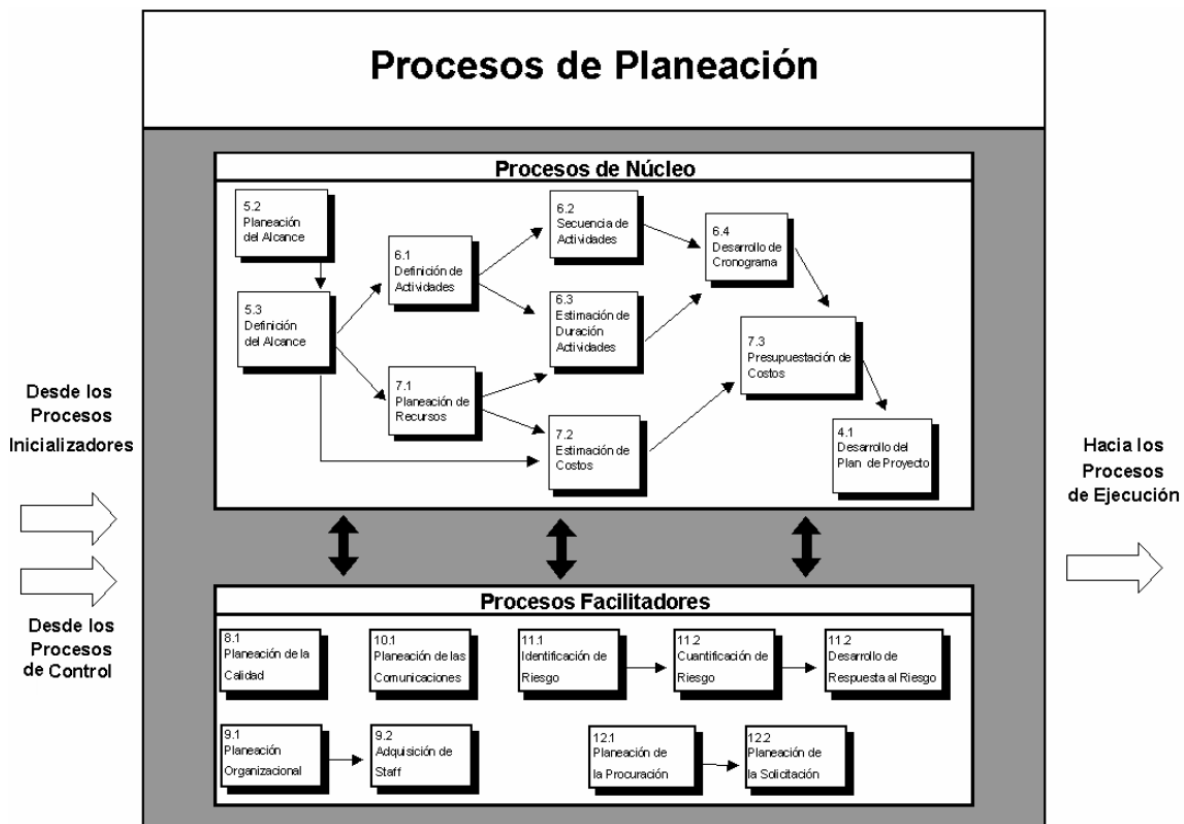


Figura 67.- Procesos de la Planeación de Proyectos

Existen dos tipos de procesos: *Procesos de Núcleo* y *Procesos Facilitadores*.

Los Procesos de Núcleo se refieren a que tienen dependencias claras con otros procesos que hacen necesario su ejecución secuencial.

Los Procesos Facilitadores en cambio no tienen una secuencia definida ni un momento exacto en la participación y tienen una alta naturaleza iterativa, ya que su ejecución puede llevarse en varios momentos de la planeación.

Dentro de los *procesos de Núcleo* tenemos los siguientes:

1. Planeación de Alcance
2. Definición de Alcance
3. Definición de Actividades
4. Planeación de Recursos
5. Secuencias de Actividades
6. Estimación de la Duración de la Actividad
7. Estimación de Costos
8. Desarrollo de la Programación
9. Presupuestos de Costos
10. Desarrollo de Plan de Proyecto.

- 1. Planeación de Alcance.** La planeación del alcance es la estrategia de cómo se obtendrán los requerimientos y se definirá el alcance del proyecto. En otras palabras, la planeación del alcance es la metodología de análisis de requerimientos para el proyecto. Todo ello debe ser documentado para así lograr el producto o servicio.

Para poder iniciar este proceso es necesario que ya exista la carta constitutiva desarrollada en la primera fase (Iniciación), ya que ahí se encuentra la descripción general de proyecto, al final del proceso de planeación de alcance se tendrán tres salidas:

1. Sentencia de alcance. Es una base documentada para una futura toma de decisiones confirmando la comprensión común del alcance del proyecto, debiendo cubrir, ya sea directa o referenciado a otros documentos, los siguientes puntos:
  - Justificación del proyecto. Se plantea la necesidad inicial por la que inicio el proyecto.
  - Refinación de las características del producto del proyecto que se describieron en el proceso anterior.
  - Lista de entregables parciales del proyecto, cuya entrega total y satisfactoria marca la terminación del proyecto.
  - Objetivos del proyecto. Criterios cuantificables que se deben cumplir para el éxito del proyecto.
2. El detalle de soporte. Se refiere a un documento donde se debe tener cuidado que no sea extenso y que sea fácil de entender para otras áreas administrativas.
3. El plan de administración del alcance. Se refiere a cómo se administrará el alcance del proyecto y cómo se documentarán los cambios.

- 2. Definición de Alcance.** En este proceso se dividen los entregables principales del proyecto, en componentes más pequeños con el fin de mejorar la precisión de los estimados de costo, duración y recursos. Una de las técnicas que se aplica en esta fase es la estructura de división de trabajo (WBS).

La descomposición en tareas (Work Breakdown Structure) es un modo de agrupar elementos del proyecto que organiza y define el conjunto del mismo. En consecuencia, no podrá existir si no hay una clara idea del alcance y los límites del proyecto y los productos a entregar o servicios a prestar. Un ejemplo de una descomposición de tareas está en la figura 68.

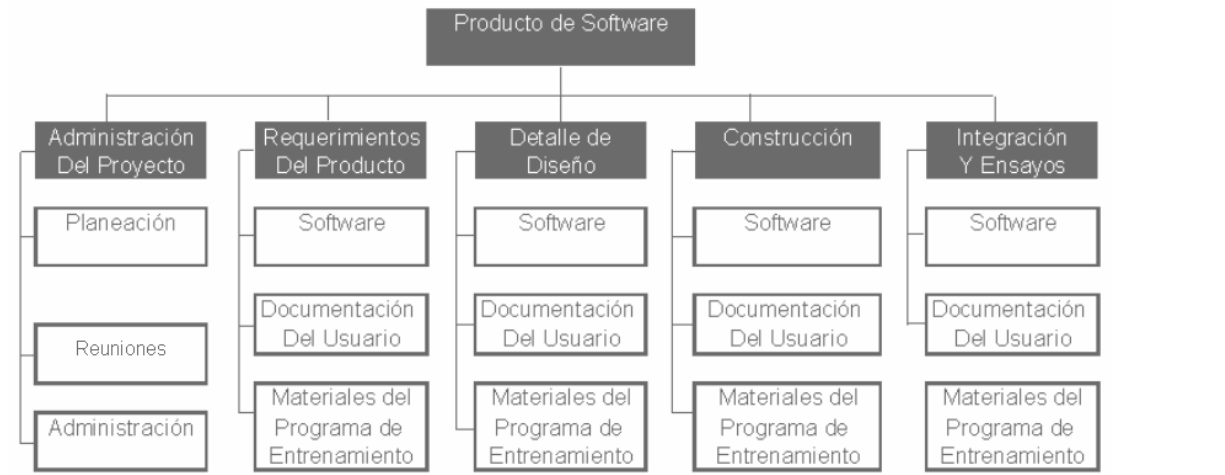


Figura 68.- Descomposición de tareas

Para lograr una buena descomposición se siguen los siguientes pasos:

1. Identificar los principales componentes del proyecto, con el fin de empezar a descomponerlos.
  2. Decidir si los estimados de costo y duración pueden ser desarrollados a este nivel de detalle. Algunos elementos manejan distintos niveles de descomposición.
  3. Si no se ha logrado un nivel de detalle aceptable, se identifican elementos constitutivos de cada entrega y describirlos en términos de resultados tangibles que ayuden a medir el rendimiento.
  4. Al final se identifica el grado de veracidad de la descomposición valorando la necesidad de los desgloses para lograr la terminación de los entregables.
- 3. Definición de Actividades.** Consiste en identificar y documentar actividades específicas que deberán ser ejecutadas para producir los entregables identificados, así como para cumplir con los objetivos del proyecto.

Una actividad que nos puede apoyar en este proceso es realizar una matriz OVAR (Objetivos Variables de Acción y Responsabilidades), en la cual se tiene un objetivo general, objetivos específicos, actividades y responsables.

4. **Planeación de Recursos.** Este proceso determinará qué recursos, personas, equipo y materiales, así como cantidades de cada uno deberán usarse para ejecutar las actividades del proyecto y la estimación de sus costos.

Para este proceso es muy importante que ya exista WBS (Estructura de Descomposición de Trabajo), ya que aquí se permite identificar los elementos de trabajo que necesitan recursos.

5. **Secuencia de Actividades.** Las secuencias de actividades es un proceso de núcleo ya que permite identificar precisamente las dependencias entre actividades. Para poder lograr hacer esta tarea existen herramientas como MS Project que automáticamente genera diagramas como el de Red si se alimenta con la información adecuada. Un ejemplo de este tipo de diagramas está en la figura 69.

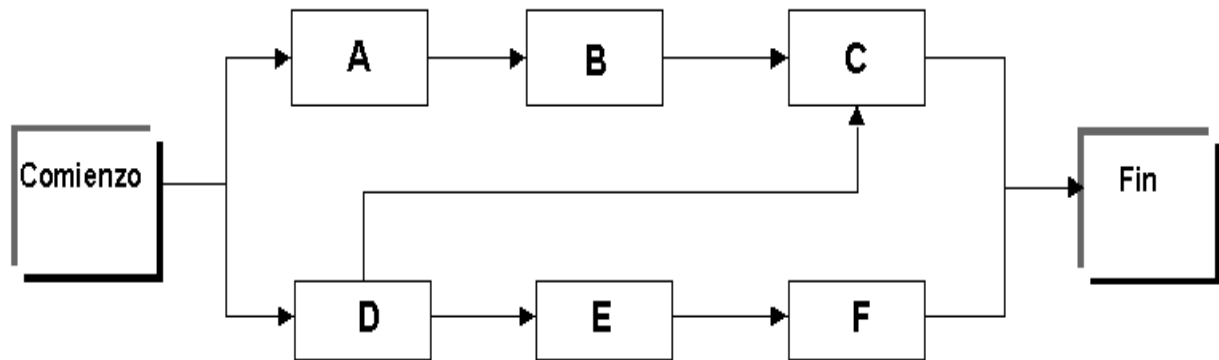


Figura 69.- Diagrama de Red

6. **Estimación de la Duración.** Este proceso tiene la finalidad de estimar el tiempo aproximado entre cada actividad. Para poder lograr esta actividad se debe considerar un calendario donde se asignen horas laborables, días de descanso, festivos, etc. Es importante considerar los recursos humanos y materiales que sin duda influenciarán la asignación de tiempos de actividades.
7. **Estimación de Costos.** En esta actividad se realiza una aproximación de los costos que tendrán los recursos requeridos para completar las actividades del proyecto. Se tiene que diferenciar que una cosa es estimar costos (lo que aproximadamente costará el proyecto) y el costeo (lo que la organización que realiza el proyecto decide cobrará).
8. **Desarrollo del Programa.** Para poder desarrollar esta actividad es necesario determinar las fechas de inicio y finalización del proyecto y es necesario tener desarrollados elementos como diagramas de red, estimación de duración de las actividades, requerimientos de recursos, recursos disponibles, restricciones, etc. Algunas herramientas que ayudan a desarrollar esta actividad son: Método de la ruta crítica, Técnicas de evaluación y revisión de programas (PERT), Diagramas de Gantt, entre otros.
9. **Presupuestos de Costos.** A diferencia de la Estimación de costos, en esta actividad se asignan dichas estimaciones a elementos de trabajo y donde el elemento importante a considerar es el tiempo, es decir, se asignan estimaciones de costos por periodos de tiempo en actividades del proyecto.
10. **Desarrollo del Plan del Proyecto.** El plan de proyecto es la principal salida de los procesos de núcleo y debe ser un documento formal, aprobado, usado para administrar y controlar la ejecución del proyecto. Para poder generar esta actividad es necesario que ya se terminen todas las anteriores actividades revisadas dentro de los procesos de Núcleo.

En la fase de Planeación tal como se indicó, existen dos tipos de procesos: los procesos de Núcleo los cuales como ya se describieron tienen dependencias claras unos de otros, y los procesos Facilitadores los cuales pueden ocurrir en diferentes partes de manera iterativa en el proyecto sin tener dependencias entre los demás. A continuación se listan los procesos Facilitadores:

- A. Planeación de la Calidad
- B. Planeación de las Comunicaciones
- C. Planeación Organizacional
- D. Adquisición de Staff
- E. Identificación del Riesgo
- F. Cuantificación del Riesgo
- G. Desarrollo de Respuesta de Riesgo
- H. Planeación de la Procuración
- I. Planeación de la Solicitación

A. **Planeación de la Calidad.** Es una actividad clave y debe ejecutarse en paralelo o regularmente con otras actividades. Para que tenga efectos positivos esta actividad debe enfocarse a la detección de posibles desviaciones y hacer una retroalimentación para la siguiente iteración. Existen algunas herramientas que nos ayudan a cumplir con esta actividad como Diagramas causa – efecto (también llamados diagramas de pescado), flujo gramas de sistemas, etc.

B. **Planeación de las Comunicaciones.** Esta actividad determina las necesidades de información y comunicaciones de los stakeholders (quién, cómo, cuándo). Es importante considerar este punto sobre todo si en el desarrollo del proyecto es vital que exista información actualizada y en línea. Además es importante considerar los recursos disponibles.

C. **Planeación Organizacional (temporales).** Así como un proyecto se realiza de manera temporal, de igual manera la planeación organización será temporal y tiene como objetivo identificar, documentar y asignar roles de proyecto. Para cumplir con esta actividad se puede hacer uso de una matriz de asignación de responsabilidades como la siguiente, en la figura 70.

FASE \ PERSONA	PERSONA						
	A	B	C	D	E	F	---
Requerimientos	S	R	A	P	P		
Funcional	S		A	P		P	
Diseño	S		R	A	I		P
Desarrollo		R	S	A		P	P
Ensayos			S	P	I	A	P

P = Participante    A= Responsable    R= Se requiere revisión  
 I= Se requiere opinión    S= Se requiere firma

Figura 70.- Matriz de Asignación de Responsabilidades

D. **Adquisición de Staff.** Esta actividad consiste en seleccionar al personal adecuado para cumplir con las tareas que el proyecto necesite, es importante hacer notar que no siempre se contará con el mejor personal, así que se tendrá que echar mano del que esté presente y garantizar que cumpla con lo que le fue encomendado. Algunos elementos a considerar para seleccionarlos son:

- Experiencia previa
- Intereses personales
- Características personales
- Disponibilidad

E. **Identificación del Riesgo.** Consiste en ubicar y documentar los posibles riesgos que hay en el proceso del proyecto. Esta actividad es iterativa, ya que en el transcurso del desarrollo del proyecto hay que estar constantemente identificando posibles riesgos. Esta actividad puede generar tres salidas:

- Riesgos, que son eventos o condiciones con incertidumbre que, si ocurren, causarán algún impacto, positivo o negativo, en el proyecto.
- Disparadores, son señales de alerta indicando un riesgo a punto de ocurrir.
- Entradas a otros procesos, estimularán acciones en otros procesos para subsanar la exposición al riesgo.

F. **Cuantificación del Riesgo.** Esta actividad tiene como objetivo determinar numéricamente o en términos de porcentaje los riesgos identificados, precisamente en el anterior punto. Se realiza un cálculo de cada riesgo y de igual forma se calcula el posible riesgo total de todo el proyecto.

G. **Desarrollo de la Respuesta de Riesgo.** Con el fin de reducir el porcentaje de riesgos en las actividades se desarrolla un plan de respuesta al riesgo, que simplemente será plantear diferentes posibilidades eliminando al máximo el posible riesgo del proyecto en su totalidad.

H. **Planeación de la Procuración.** Es el proceso de identificar las necesidades del proyecto que pueden ser mejor cumplidas al adquirir productos o servicios externos a la organización. Esto involucra considerar si hay que adquirir, cómo adquirir, cuánto adquirir y cuándo adquirirlo.

I. **Planeación de la Solicitación.** Se refiere a los documentos necesarios para poder solicitar lo necesario para el desarrollo de las actividades del proyecto.

Después de la recolección de información de la fase de Iniciación y la Planeación, a través de diversas herramientas, se pasará a la siguiente fase, a la que se refiere a iniciar la construcción física de todo el proyecto: La Ejecución.

## **7.9 Ejecución**

En esta fase se coordina a las personas y otros recursos de acuerdo a los planes establecidos en la fase anterior. De la misma manera que en la fase de planeación en esta fase existen dos tipos de procesos: de núcleo y facilitadores como lo muestra la figura 71.



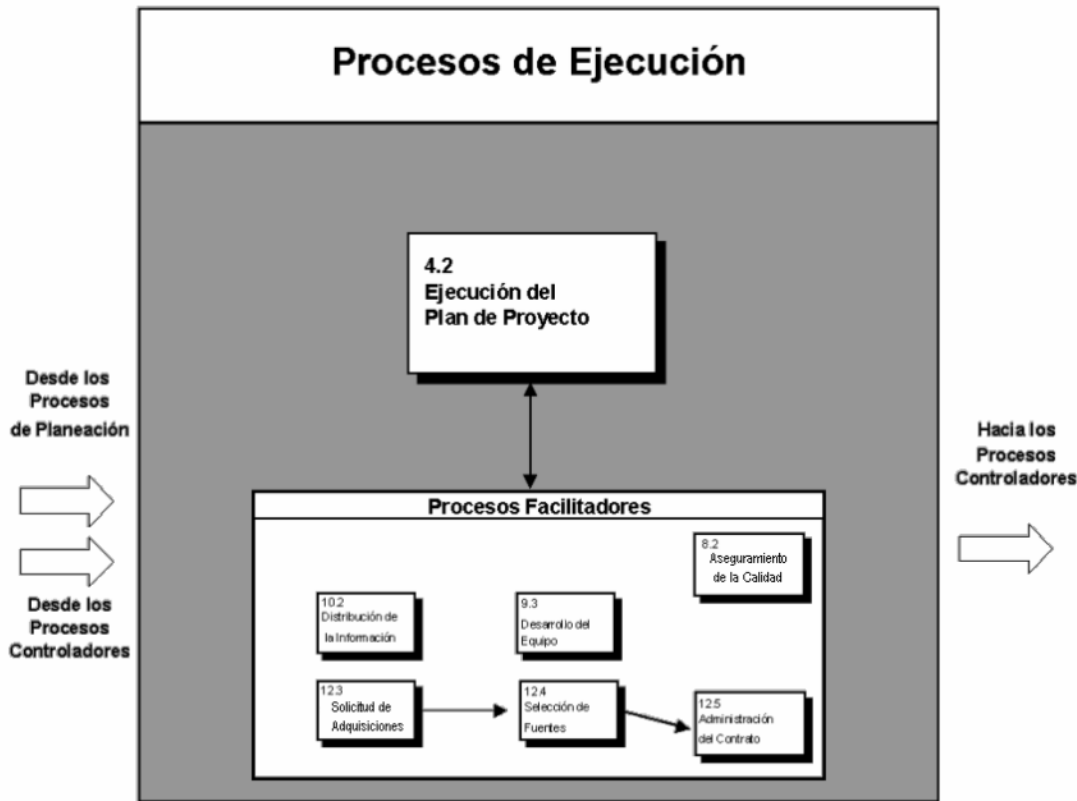


Figura 71.- Procesos de Ejecución del Proyecto

Dentro de los procesos de Núcleo se tienen los siguientes:

#### Ejecución del Proyecto

1. Diseño de la solución TI
2. Preparación del ambiente de desarrollo
3. Desarrollo de la solución de TI
4. Pruebas y certificación del alcance
5. Instrumentación de la solución
6. Capacitación
7. Proceso de liberación, monitoreo y soporte técnico

Es en este proceso donde se lleva a cabo el plan de proyecto diseñado en la anterior fase, este proceso primario tiene la finalidad de crear el proyecto o producto en sí. Para lograr esta actividad es necesario que exista el plan de proyecto, manejo de riesgo, administración de alcance entre otros elementos. Al final de este proceso se tienen por ejemplo: solicitudes de cambio sobre su alcance y lo más importante un entregable que se irá puliendo con el paso de iteraciones.

1. **Diseño de la solución de TI.** Para lograr esta actividad es importante apoyarse de los requerimientos FURPS+ ya planteados al inicio del proyecto, donde precisamente se tiene la identificación de los requerimientos del producto o servicio.

Cuando se inicia un proyecto, el primer objetivo es que sea autorizado y para esto hay que modelarlo, y como ya revisó, existen herramientas de desarrollo para este fin, UML (Lenguaje de Modelado Unificado) es una herramienta que aportará nuevamente en esta fase del proyecto.

Algunos elementos que deben ser definidos son los siguientes:

- Herramientas de desarrollo, incluyendo versiones que se manejarán
- Protocolos de comunicación
- Nomenclatura de elementos (variables funciones, interfaces, paquetes, tablas, atributos, etc.)
- Dominios de los tipos de datos
- Documentación técnica como manejo de colores, distribución y tipos de elementos, etc.

En seguida se muestra en las figuras 72 y 73 como modelar el diseño del Proyecto. La figura 72, muestra Modelo Lógico donde se pueden aterrizar varios elementos ya mencionados en la fase de planeación y la figura 73 es más detallada, ya que se visualizan los tipos de datos, relaciones entre ellos e incluso el programa en que se utilizará.

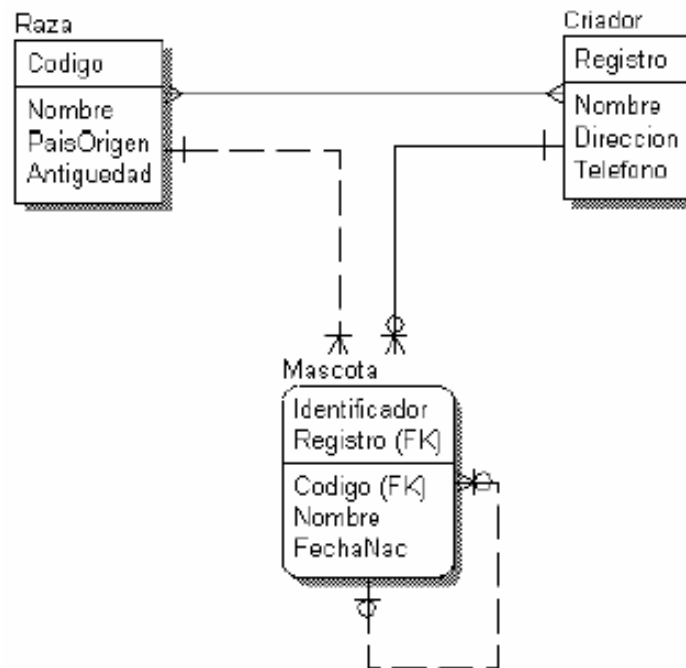


Figura 72.- Modelo Lógico

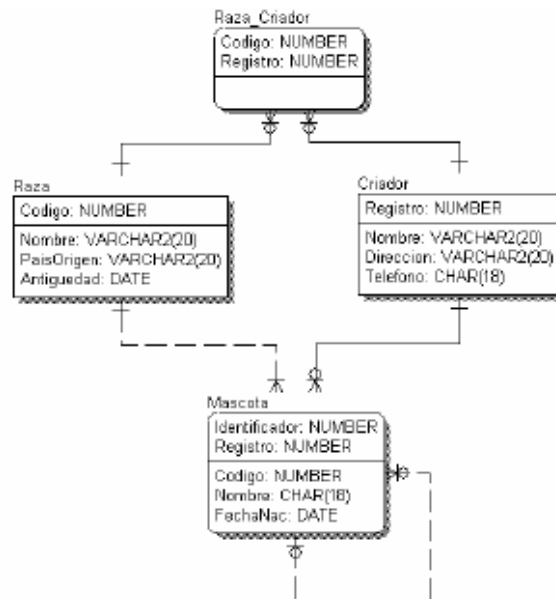


Figura 73.- Modelo Físico en Oracle

2. **Preparación del ambiente de desarrollo.** En este proceso se identifican los requerimientos de desarrollo, como por ejemplo: recursos físicos y económicos, programas, licencias, bases de datos, etc.
3. **Desarrollo de la solución de TI.** En esta parte del proyecto es donde el analista junto con los desarrolladores construyen la solución planteada, y tal como debe imaginarse es aquí donde los desarrolladores juegan un papel sumamente importante, ya que de ellos depende la construcción de los módulos de software si se tratara el proyecto de un software o de la aplicación de instrumentación tecnológica adquirida si se tratase de hardware.
4. **Pruebas y certificación del alcance.** Una vez que se desarrolló una solución el siguiente paso es someterle a pruebas antes de liberarlo en un ambiente de producción. Para lograr esta actividad se realizan sesiones con los usuarios que utilizarán el sistema y es aquí donde se encuentran barreras como la resistencia al cambio tecnológico que experimentan dichos usuarios.

Otro elemento importante a considerar en este proceso es que debe llevarse a cabo una bitácora donde se anoten los posibles errores que arroja la solución para poder retroalimentar las iteraciones siguientes. Al final de este proceso debe lograrse la satisfacción del usuario sin errores en la solución.

5. **Instrumentación de la solución.** Este proceso tiene la finalidad de instalar, configurar y verificar el funcionamiento correcto de la solución de TI desarrollada, así como la migración planeada de la solución vieja a la nueva, verificando que no se afecte la continuidad de la operación.

Otro elemento importante que se desarrolla en este proceso es la creación de un manual de instalación y configuración de la herramienta. Al final, se debe lograr la solución implantada funcionando correctamente y logrando la aceptación del usuario.

6. **Capacitación.** En este proceso se realizan roles para enfrentar a los usuarios con el sistema creado, la mejor manera de acercar al usuario con la aplicación es la práctica, es importante hacer notar que no es suficiente con ofrecer un manual de usuario ya que si el usuario no le convence, el sistema no rechazará y buscará soluciones por su cuenta sin utilizar el sistema lo que provocara un virtual fracaso.

Al final de está actividad se realizan evaluaciones a los usuarios verificando el aprovechamiento y el convencimiento de los mismos.

7. **Proceso de liberación, monitoreo y soporte técnico.** Al final de todo el proceso se debe estar monitoreando la herramienta creada, ya que de este proceso depende que se cumplan con las expectativas planteadas. A continuación se muestran 2 gráficas en las figuras 74 y 75, donde se puede observar la diferencia de la curva de aprendizaje que tienen los usuarios sin soporte técnico y con soporte técnico.

Como puede observarse en la figura 75, la curva de aprendizaje tiene una caída menor, ya que existe el apoyo de soporte técnico para la correcta adaptación de los usuarios.

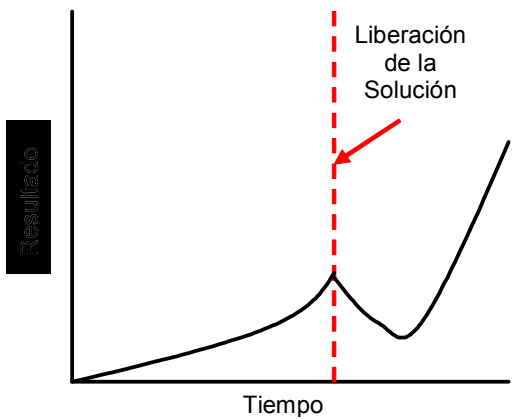


Figura 74.- Curva Típica de Aprendizaje

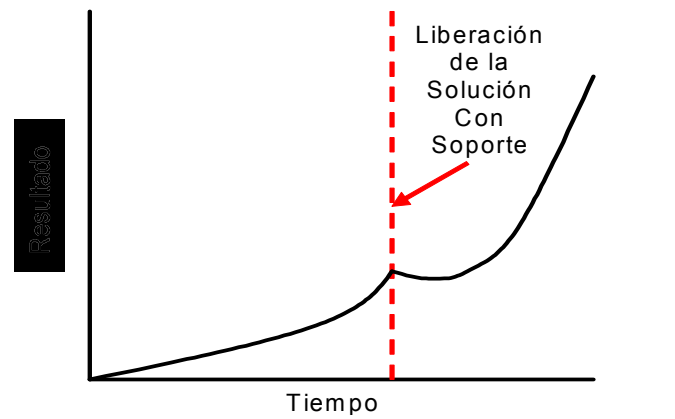


Figura 75.- Curva de Aprendizaje con Soporte Técnico

En la misma fase de ejecución se tienen procesos facilitadores que pueden ocurrir en diferentes iteraciones en los diferentes procesos de núcleo vistos anteriormente. Los procesos facilitadores de esta fase se pueden ver en la figura 76.

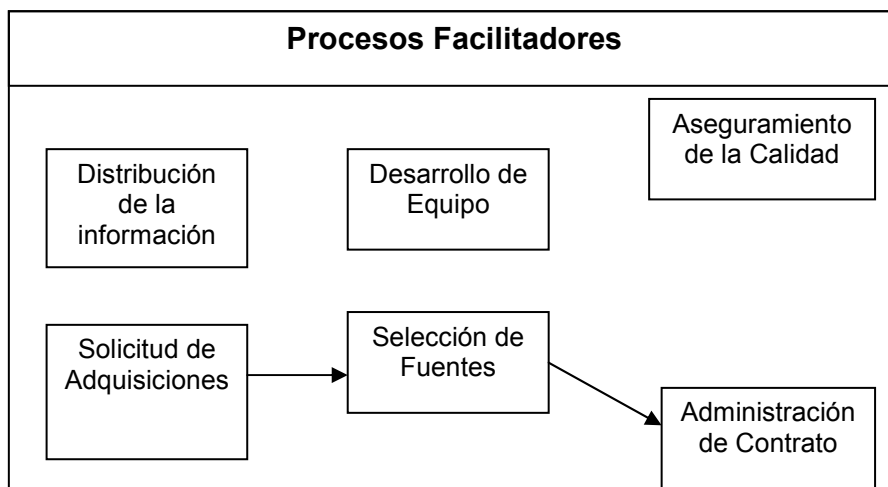


Figura 76.- Procesos Facilitadores

En seguida se revisará cada uno de estos procesos facilitadores, explicando a qué se refiere cada uno de ellos:

- **Distribución de la Información.** Esta actividad tiene la finalidad de garantizar que la información esté disponible en forma oportuna por los individuos que la requieran. Un elemento importante es que hay que tener respaldos de la información.
- **Desarrollo de Equipo.** Este proceso enfoca el mejoramiento de las habilidades de los miembros del equipo de trabajo, para enriquecer su labor individual y asegurar que el equipo funcione como equipo.
- **Aseguramiento de Calidad.** Este proceso facilitador tiene que auditar constantemente los procesos antes descritos para lograr detectar posibles desviaciones y poder corregirlos.
- **Solicitud de Adquisiciones.** Consiste en obtener información de licitadores y propuestas de los proveedores potenciales sobre como las necesidades del proyecto se pueden cumplir. La mayor parte del esfuerzo de este proceso es realizado por los proveedores, normalmente sin costo para el proyecto.
- **Selección de Fuentes.** En el proceso anterior tal cómo se explicó se recaba información de los posibles proveedores, en este proceso se realiza la selección dependiendo de las necesidades del proyecto.
- **Administración de Contrato.** La administración de contratos es el proceso de asegurar que el desempeño del vendedor cumplirá con los requerimientos conceptuales.

Se debe recordar que existen iteraciones que permiten estar detectando posibles desviaciones o corrección total del proyecto. Para precisamente evitar esto, existe una cuarta fase llamada de Control. En seguida se estudiará a qué se refiere esta última fase.

### 7.10 Control

Esta fase del proyecto se encarga de la constante verificación del proyecto ya ejecutado e instalado. El objetivo es cumplir con las metas establecidas desde un inicio, si se encuentran variaciones se toman acciones para encaminar nuevamente el proyecto.

La fase de control al igual que las anteriores fases cuenta con procesos de núcleo y procesos facilitadores, los cuales se revisarán uno a uno y gráficamente se pueden ver en la figura 77.

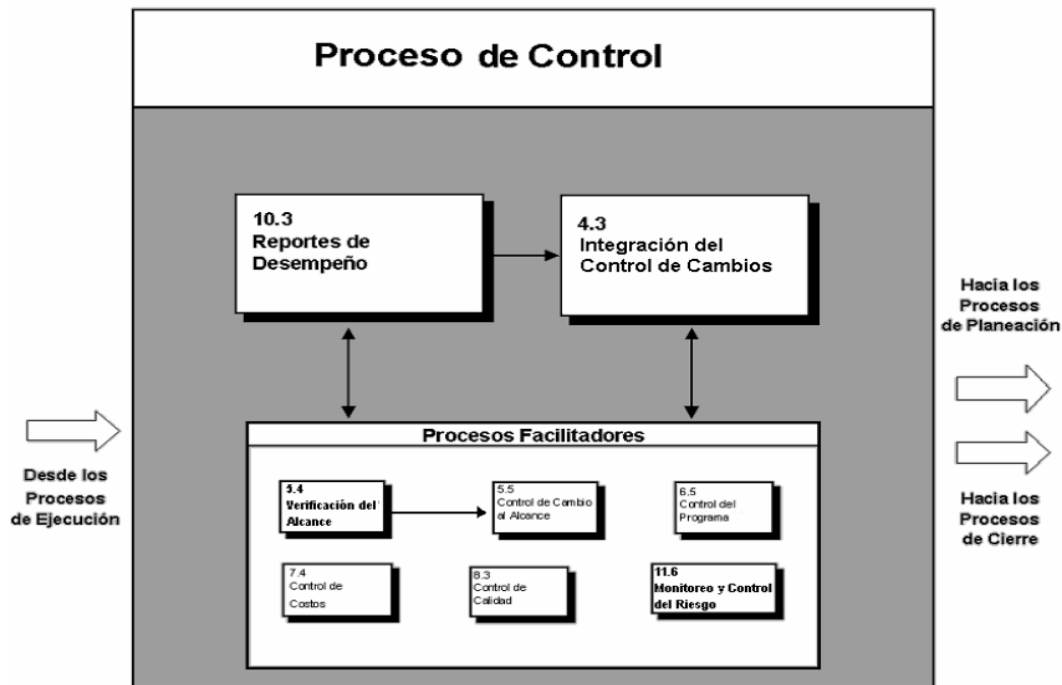


Figura 77.- Procesos de Control

Dentro de los procesos de Núcleo se tiene:

- **Reportes de Desempeño.** Este proceso mide mediante el uso de herramientas el desempeño del proyecto, así como los recursos que se están utilizando y que realmente tengan un uso adecuado. Algunas herramientas son:
  - Reportes de Estado
  - Reportes de progreso, sólo los productos terminados
  - Pronósticos

Al final, el reporte de desempeño es presentado a los stakeholders para que puedan evaluar si todo marcha bien.

- **Integración de control de cambios.** Este proceso tiene como objetivo documentar e integrar los cambios realizados al proyecto garantizando que efectivamente se lleven a cabo dichos cambios, además de asegurar que sean benéficos para la empresa. Dichos cambios deberán ser informados a los stakeholders.

Los procesos Facilitadores son los siguientes:

- **Verificación de alcance.** Este proceso busca la aceptación formal del producto a través de la documentación en la que el cliente o patrocinador avale que acepta el producto del proyecto o fase.
- **Control de cambios de alcance.** En esta actividad se definen los procedimientos a través de los cuales el proyecto será cambiado, incluyendo formatos, sistemas de seguimiento y niveles de aprobación para autorizar el cambio.

- **Control del programa.** Este proceso tiene como objetivo obtener beneficios de los factores que generan los cambios en la programación del proyecto, identificando fechas de cambio de programación administrando sucesos como la forma en que ocurren.
- **Control de costos.** Este proceso tiene la finalidad de garantizar que los costos estimados hayan sido los proyectados en base al plan inicial, al final se pretende estimar los costos revisados, dentro de estos colocar los cambios realizados, además de notificar a los stakeholders.
- **Control de calidad.** El control de calidad deberá ejecutarse durante todo el proyecto y sus resultados deben incluir tanto resultados del producto (entregables parciales) como resultados administrativos (desempeño de costos y programación).
- **Monitoreo y control de riesgos.** Es el proceso que da seguimiento a los riesgos identificados, monitorear los riesgos residuales e identificar nuevos riesgos, asegurando la ejecución de los planes de respuesta al riesgo y evaluando su efectividad.

Como proceso final y última fase, se realiza un cierre, es decir, finalizar todos los procesos y tareas puestas en marcha.

### 7.11 Cierre

La última fase de la creación de un proyecto de tecnologías de TI se refiere al Cierre y tiene el objetivo de formalizar la aceptación del proyecto, fase o entregable llevando a una terminación ordenada. El cierre del contrato completa y negocia un contrato, incluyendo la resolución de cualquier elemento abierto o pendiente.

Esta Fase tiene 2 procesos. Gráficamente se puede ver en la figura 78:

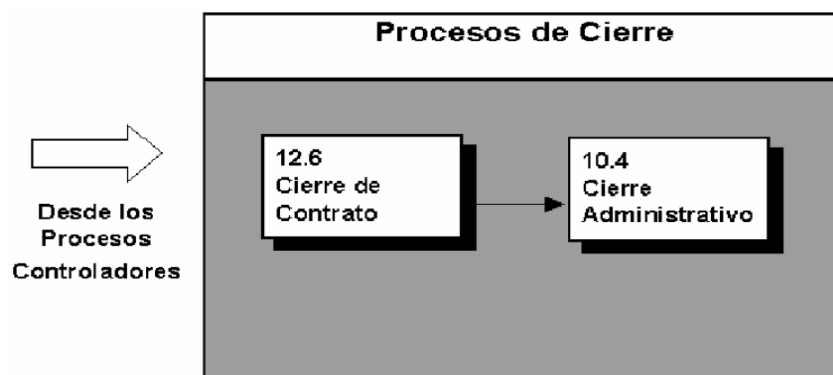


Figura 78.- Procesos de Cierre

Enseguida se revisará a que se refiere cada una de ellas:

**Cierre de Contratos.** En este proceso se realiza una verificación del producto, términos y condiciones que señalan procedimientos específicos para el cierre del contrato. Lo más importante es lograr la aceptación formal del cierre del proyecto.

**Cierre Administrativo.** Al final el cierre administrativo se encarga de recoger, generar y repartir la información para la formalización del cierre del proyecto.

Estos dos procesos finales concluyen todo el proceso de la generación de un proyecto. Desde luego que la creación de un proyecto de proporciones considerables tal como se revisó, si se quiere que sea un sistema con alta calidad, aportando conocimientos de redes de la misma computadora, las herramientas existentes de apoyo, las diferentes aplicaciones y las diferentes formas de hacerlo, debe considerarse todo lo estudiado en el presente Diplomado y todo con el objetivo de preservar la información, transmitirla, aportando conocimiento a las siguientes generaciones.



## Capítulo 8

### Resultados y Conclusiones

La información es fundamental dentro de una organización, es tema esencial para poder ser exitosa y sobresalir, las nuevas tecnologías de información (TI) han desarrollado posibilidades para almacenarlas y manejarlas en las organizaciones, las empresas que logren aprovechar estas tecnologías y mejorar sus procesos de negocio, tendrán ventajas competitivas y estratégicas de vital importancia.

En la actualidad se puede obtener la información de un gran número de fuentes diversas, sin embargo, muchas empresas todavía no han adoptado las Tecnologías de Información como una herramienta básica para su desarrollo y competencia. Las Tecnologías de Información pueden ayudar a mejorar la productividad de todas las funciones de la empresa, y el flujo de información dentro y fuera de un negocio. Una organización que pretenda ser efectiva deberá de explotar y administrar todas estas tecnologías.

El presente trabajo, por lo tanto, aporta a los profesionistas, ingenieros, administradores o a todo aquel que haga uso de tecnologías de información en las organizaciones, el conocimiento y uso de algunas de éstas, mostrando un panorama amplio desde los conceptos básicos de la computación, hasta las aplicaciones reales en el mundo de las organizaciones.

Las tecnologías de información permiten a una empresa manejar sus necesidades de información, las grandes personalidades de una empresa pasan grandes cantidades de tiempo buscando soluciones para cada problema que se presenta y es ahí cuando las tecnologías de información cumplen un papel muy importante para la resolución de éstos.

Algo esencial para una empresa es el dinero y el tiempo, mediante las Tecnologías de Información podemos reducir los costos empleados para hacer algo y esto hace que se pueda mejorar la productividad de la empresa en todos sus ámbitos.

Desde luego, las empresas a fin de cuentas buscan crecimiento y lograr mejores ganancias, es por ello que el estar a la vanguardia les ofrece ventajas sobre sus competidores y para lograr este cometido, es necesario estar en constante actualización de capacitación de su personal.

El incremento en el uso del correo electrónico, el Internet, y el desarrollo de Intranets o redes de comunicaciones entre empresas, está acelerando el flujo de información en las empresas y negocios. Todos estos sistemas de transferencia y recuperación de información están basados en el uso de redes y computadoras personales unidas unas con otras y todas conectadas a una computadora central que permite a los usuarios compartir archivos e información digital de todo tipo.

La revolución de las Tecnologías de Información ha tenido un profundo efecto en la administración de las organizaciones, mejorando la habilidad de los administradores para coordinar y controlar las actividades de la organización y ayudándolos a tomar decisiones mucho más efectivas.

Al final el presente trabajo contribuye, a que el personal interesado del área de la informática, esté interesado en la constante actualización y capacitación reforzando los conocimientos que día con día contribuyen al fortalecimiento de las instituciones, y el aprovechamiento y superación personal. En lo particular al estar inmerso en el medio, desde luego que aporta para mi desarrollo profesional ya que me brinda un panorama amplio de lo que actualmente se esta demandando.

### Referencias

- [1]. <http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf>
- [2]. [http://es.wikipedia.org/wiki/John\\_von\\_Neumann](http://es.wikipedia.org/wiki/John_von_Neumann)
- [3]. <http://redescolar.ilce.edu.mx/redescolar/cursos/sepacomputo/intrcomp.pdf>
- [4]. <http://iteso.mx/~jluis/acpdf/oto-04/02-Evolucion-de-las-computadoras.PDF>
- [5] <http://www.monografias.com/trabajos41/sistema-numeracion/sistema-numeracion2.shtml>
- [6] [http://www.lookuptables.com/ebcdic\\_scancodes.php](http://www.lookuptables.com/ebcdic_scancodes.php)
- [7] [http://almacen.gulic.org/diveintopython-5.4-es/xml\\_processing/unicode.html](http://almacen.gulic.org/diveintopython-5.4-es/xml_processing/unicode.html)
- [8] <http://www.itz.edu.mx/~lveyna/ensamblador/unidad1.htm>
- [9] <http://www.geocities.com/hectorabrahan/SO/historia.htm>
- [10] <http://www.monografias.com/trabajos19/sistemas-operativos/sistemas-operativos.shtml>
- [11] <http://www.monografias.com/trabajos13/modosi/modosi.shtml>
- [12] <http://html.rincondelvago.com/medios-de-transmision.html>
- [13] <http://www.mailxmail.com/curso/informatica/wifi>
- [14] <http://www.eveliux.com/articulos/estandareswlan.html>
- [15] [http://www.unicrom.com/Tel\\_RF2.asp](http://www.unicrom.com/Tel_RF2.asp)
- [16] <http://ditec.um.es/laso/docs/tut-tcpip/>
- [17] <http://www.siste.com.ar/modulacion.htm>
- [18] [http://manuscritovoynich.blogspot.com/2005\\_06\\_12\\_manuscritovoynich\\_archive.html](http://manuscritovoynich.blogspot.com/2005_06_12_manuscritovoynich_archive.html)
- [19] [http://people.debian.org/~sto/articulos\\_bei/BEI-2003-04/criptologia.html](http://people.debian.org/~sto/articulos_bei/BEI-2003-04/criptologia.html)
- [20] [http://www.terra.es/personal6/morenocerro2/seguridad/cripto/cripto\\_5.html](http://www.terra.es/personal6/morenocerro2/seguridad/cripto/cripto_5.html)
- [21] <http://sistemas.itlp.edu.mx/tutoriales/basedat1/temas1.htm>
- [22] <http://www.cs.cinvestav.mx/BDChapa/Beto/Blanco.htm>
- [23] <http://www.manycomics.com/ingenieria-del-software/ciclo-vida-software/>
- [24] <http://www.javahispano.org/articles.article.action?id=76>
- [25] <http://www.programacion.com/tutorial/uml/>
- [26] <http://www.desarrolloweb.com/articulos/2358.php>
- [27] <http://www.monografias.com/trabajos20/paradigmas-de-programacion/paradigmas-de-programacion.shtml>
- [28] <http://mena.com.mx/gonzalo/maestria/ingsoft/presenta/rad/>
- [29] [http://www.cad.com.mx/historia\\_del\\_internet.htm](http://www.cad.com.mx/historia_del_internet.htm)
- [30] [http://www.utem.cl/web/urlhtml\\_fm.htm](http://www.utem.cl/web/urlhtml_fm.htm)
- [31] <http://html.conclase.net/tutorial/html/>
- [32] <http://www.ulpgc.es/otros/tutoriales/xml/>
- [33] <http://www.itapizaco.edu.mx/paginas/JavaTut/froufe/introduccion/indice.html>
- [34] [http://www.asturlinux.org/archivos/jornadas2004/ponencias/eclipse\\_ide.pdf](http://www.asturlinux.org/archivos/jornadas2004/ponencias/eclipse_ide.pdf)
- [35] <http://dis.um.es/~jfernand/0405/smig/tema1.pdf>
- [36] <http://www.abcdatos.com/tutoriales/tutorial/16345.html>