

**PAGINACION VARIA**

**TESIS SIN PAGINACION**

20  
29



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**FACULTAD DE CONTADURIA Y ADMINISTRACION**

**EL USUARIO Y LA CONSTRUCCION  
DE UN SISTEMA**

**SEMINARIO DE INVESTIGACION EN INFORMATICA**

QUE PARA OBTENER EL TITULO DE:  
LICENCIADO EN INFORMATICA

**P R E S E N T A :**  
**FRANCISCO JAVIER VILLEGAS LANDIN**

*ASESOR:*

*C. P. Y L. A. JOSE ANTONIO ECHENIQUE GARCIA*

**TESIS CON  
FALLA DE ORIGEN**

**1994**



Universidad Nacional  
Autónoma de México



## **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 está 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.

**PAGINACION VARIA**

**TESIS SIN PAGINACION**



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**FACULTAD DE CONTADURIA Y ADMINISTRACION**

**EL USUARIO Y LA CONSTRUCCION  
DE UN SISTEMA**

**SEMINARIO DE INVESTIGACION EN INFORMATICA**

QUE PARA OBTENER EL TITULO DE:  
LICENCIADO EN INFORMATICA

**P R E S E N T A :**

**FRANCISCO JAVIER VILLEGAS LANDIN**

*ASESOR:*

*C. P. Y L. A. JOSE ANTONIO ECHENIQUE GARCIA*

**TESIS CON  
FALLA DE ORIGEN**

**1994**

## **Dedicatoria y Agradecimientos**

En los momentos que escribo éste párrafo vienen a mi mente cientos de personas que me han brindado su apoyo para superarme en todos los aspectos de mi vida. A ellos deseo darles las gracias.

El principal anhelo que he tenido en mi vida es dar las gracias a todas aquellas personas que me han regalado una sonrisa simplemente por el hecho de estar vivo.

Muy especialmente deseo dar las gracias y dedicar el presente documento a mis Padres: Sra. María de los Angeles Landín Ibarra Vda. de Villegas y al Sr. Francisco Villegas Espinosa, -que aunque no se encuentra de cuerpo presente-, dieron lo que fueron y han podido ser para que yo saliera adelante.

Otra persona a la cual deseo dedicar el Seminario de Investigación es a mi esposa Blanca Graciela. López Jiménez quien me ha brindado su apoyo y amor, así como también a mi hija Arantxa Berenice.

También deseo brindar la presente a mis sobrinos Erick Mauricio, Luis Edgar y Cinthya Penelope.

*A todos ustedes muchas gracias.*

## Introducción

Durante los últimos años los desarrolladores de sistemas, como en algunas metodologías se indica, hemos usado como única herramienta para definir las necesidades de información de los usuarios a la entrevista informal.

Llevar a cabo ésta técnica depende en mucho de la interpretación que se le de a las diferentes literaturas de metodología de sistemas. Algunos desarrolladores o ingenieros de sistemas pensamos que para el inicio del desarrollo de un sistema *sólo basta con una simple entrevista*, en la cual el usuario determine sus requerimientos; quedando así bajo la responsabilidad del ingeniero de sistemas la concepción y la construcción del sistema.

El usuario explica al ingeniero de sistemas la serie de necesidades de información que identificó *sin hacer un análisis detallado de su negocio*. Esto como consecuencia crea una dependencia de la interpretación que le de el ingeniero de sistemas, ya que de la concepción que tenga el usuario de su negocio y de lo que transmita al ingeniero de sistemas dependerán los resultados y productos que genere el futuro sistema.

El usuario es el *experto del negocio* y como tal conoce su función, procesos y requerimientos. La función del ingeniero de sistemas es la automatización de la información, *él no es el experto del negocio*.

Personalmente, basado en mi experiencia, éste tipo de concepción del desarrollo de un sistema no ha sido satisfactoria. Considero que la relación entre el usuario y el ingeniero de sistemas debe ser de una *gran participación y alta interacción*, ya que el ingeniero de sistemas construirá un sistema que generará los resultados tal y como él los percibió en el análisis que haya realizado.

Pienso que se requiere de una *estrecha interacción* del usuario y el ingeniero de sistemas, para que el resultado sea un sistema que cumpla al cien por ciento con los requerimientos del Usuario.

*El objetivo de este trabajo*, es proponer y dar a conocer al usuario una serie de técnicas y herramientas integradas de tal forma que le pueden ayudar para participar en el proceso de la construcción de un sistema.

Todo el documento pretende mirar el proceso de construcción de un sistema desde el punto de vista del usuario, sin pretender convertir a éste en un experto en automatización de procesos de información; pero si dando un apoyo para su conceptualización.

El capítulo da a conocer al lector una serie de conceptos que son trabajados durante todo el documento y que además desde un muy particular punto de vista debe conocer para la interacción con los ingenieros de sistemas.

El capítulo dos describe los componentes de un sistema de información, así como ubicar al usuario en diferentes categorías; Así mismo se describe la importancia de la participación del usuario en el proceso de construcción; posteriormente aborda el tema de Planeación Estratégica de Sistemas haciendo énfasis en los beneficios que genera su utilización.

El capítulo tres describe los pasos para elaborar la auditoría operacional como fase inicial para el desarrollo de un sistema, posteriormente se hace una descripción detallada de SADT, y es entonces cuando se habla la técnica para elaborar un diseño conceptual para finalizar con la descripción para hacer la prueba operativa del sistema de información que se ha construido.

El capítulo cuatro muestra una serie de técnicas de apoyo para la continuación del proceso de construcción, se toca el tema de administración de datos, la nomenclatura y la seguridad de los mismos; así mismo se explica que es la técnica de entidad relación llave atributo y que es un diccionario de datos.

## **Auditorio hacia quien va dirigido el Seminario**

El presente Seminario de Investigación está dirigido a los usuarios de los sistemas de información y a aquellas personas que participen en el desarrollo de los mismos.

El Seminario es enunciativo, pretendiendo dar elementos básicos a los usuarios acerca del proceso de construcción de un sistema de información en el *negocio*<sup>1</sup>.

Contiene información que personalmente considero que los usuarios deben conocer para interactuar con el área de ingeniería de sistemas.

Es muy importante recordar que "*el negocio*" podrá ser cualquier área del conocimiento humano, por lo tanto el presente documento esta orientado a cualquier persona que sea usuario de un sistema de información.

---

<sup>1</sup> Al referirme al negocio quiero decir a la organización.

## Indice

Introducción.....	III
Auditorio hacia quien va dirigido el Seminario .....	V
Indice .....	VI
CAPITULO I .....	11
Conceptos Generales .....	11
<b>PARTE I. DE CONCEPTO Y DISEÑO.</b> .....	12
Entidad.....	12
Atributo de una Entidad. ....	12
Ocurrencia. ....	12
Relación.....	12
Llave.....	12
Llave Primaria. ....	13
Llave Alterna. ....	13
Reglas de Integridad. ....	13
Reglas del Negocio.....	14
Modelo de Información. ....	14
Requerimiento . ....	14
Software.....	15
Hardware. ....	15
<b>PARTE II. DE ESTRUCTURA Y ALMACENAMIENTO.</b> .....	16
Bit .....	16
Ascii.....	17
Ebdic.....	17
Byte .....	19
Equivalencias .....	19
Caracter.....	19
Palabra o Cadena. ....	20
Campo. ....	20
Registro. ....	20
Dato. ....	20
Estructura de Datos .....	20
Información. ....	20
Archivo.....	20
Base de Datos .....	21
Sistema Administrador de Base de Datos (DBMS).....	22
Tabla.....	23
Índices.....	23
Vistas.....	24

Bases de Datos vs. Archivos.....	25
Apuntador .....	26
Listas.....	26
Arreglos.....	27
Pilas.....	30
Colas.....	31
Una Breve Historia de la Computación.....	31
Computadora.....	32
BootStrap. (Autoinicialización).....	33
Memoria.....	33
RAM .....	33
ROM.....	34
Disco Duro .....	34
Disco Flexible.....	34
Sistema Operativo.....	35
Flujo General de Operación de una Computadora.....	35
La Unidad Central de Proceso (UCP/CPU).....	36
<b>PARTE III. DE PROCESAMIENTO.....</b>	<b>38</b>
Proceso.....	38
Mensaje.....	38
Programa.....	38
Programa Fuente.....	39
Compilador.....	39
Programa Objeto.....	39
Proceso de Ligado.....	39
Librerías.....	40
Programa Ejecutable.....	40
Intérprete.....	40
Transacción.....	40
Bitácora de Transacciones.....	41
Proceso Local.....	41
Proceso Remoto.....	42
Proceso Batch .....	42
Proceso en Línea .....	43
Proceso en Tiempo Real.....	43
Proceso Distribuido.....	44
<b>PARTE IV. PLATAFORMAS.....</b>	<b>46</b>
Las Micro Computadoras o Computadoras Personales.....	46
Como una PC Stand Alone (Mono Usuario) .....	46
Como un Servidor de Archivos .....	46
Como una Estación de Trabajo.....	47
Como una Terminal Inteligente .....	47
Mainframe.....	48

Host .....	48
Terminal .....	48
Red de Area Local .....	48
Red de Redes .....	48
<b>CAPITULO II</b> .....	49
<b>Los Sistemas de Información</b> .....	49
<b>PARTE I. COMPONENTES DE UN SISTEMA DE INFORMACIÓN</b> .....	50
El Sistema .....	50
El Insumo .....	50
El proceso .....	50
El producto .....	51
El Usuario .....	51
Una Clasificación de los Usuarios .....	52
El Ingeniero de Sistemas .....	53
<b>PARTE II. LA PARTICIPACIÓN DEL USUARIO</b> .....	54
<b>PARTE III. PLANEACION ESTRATEGICA DE SISTEMAS (BSP)</b> .....	58
Objetivos de la Planeación Estratégica de Sistemas .....	59
Beneficios Potenciales .....	59
<b>PARTE IV. CONCEPTOS BSP</b> .....	61
Proceso del Estudio BSP .....	62
1. Compromiso .....	62
2. Preparación del Estudio .....	62
3. Inicio del Estudio .....	62
4. Definición de los Procesos del Negocio .....	62
5. Definición de los Datos del Negocio .....	62
6. Definición de la Arquitectura de la Información .....	63
7. Análisis del Soporte a los Sistemas Actuales .....	63
8. Entrevistas a los Ejecutivos .....	63
9. Conclusiones .....	64
10. Definición de las Prioridades .....	64
11. Desarrollo de Recomendaciones .....	64
12. Compromiso .....	65
<b>CAPITULO III</b> .....	66
El Proceso de Construcción .....	66
<b>PARTE I. AUDITORIA OPERACIONAL</b> .....	67
1. Establecer el Alcance, Calendario y Presupuesto del Proyecto .....	67
2. Identificar los Procesos del Sistema a Desarrollar .....	68
3. Identificar que actividades se realizan y en qué departamentos .....	68
4. Preparar las Entrevistas con el Usuario .....	69
5. Documentación .....	69
6. Proceso de Análisis y Síntesis .....	69
7. Construcción del Macro Modelo de Información .....	70
8. Construcción de Modelos de Actividades para cada Proceso .....	70

9. Producción del Reporte Final .....	70
<b>PARTE II. TECNICA DE ANALISIS Y DISEÑO ESTRUCTURADO.....</b>	<b>71</b>
<b>STRUCTURED ANALYSIS DESIGN TECHNIQUE "SADT".....</b>	<b>71</b>
Túnel SADT.....	79
La diferencia entre Entrada y Control.....	79
Los Mecanismos definen como las actividades son realizadas.....	79
Retroalimentación de Control y Flujo de Datos.....	80
El Lenguaje Natural en SADT.....	80
El proceso de Descomposición.....	80
Descomposición Funcional.....	81
Descomposición por Rol del Personal.....	81
La precisión determina cuando se detiene la descomposición.....	81
<b>PARTE III. TECNICA PARA EL DISEÑO CONCEPTUAL.....</b>	<b>82</b>
1. Desarrollo de las Vistas del Usuario.....	82
2. Refinamiento del Modelo de Información.....	82
3. Identificar los Requerimientos de Entrada.....	83
4. Identificar los Requerimientos de Aplicación.....	84
5. Construir la Matriz de Desarrollo.....	85
6. Producir el Reporte Final.....	86
Productos a Obtener al Final de la Fase del Diseño Lógico.....	86
<b>PARTE IV. PRUEBA OPERATIVA.....</b>	<b>88</b>
1. Preparar el Plan de la Prueba Operativa.....	88
2. Llevar a cabo pruebas de Integración de Sistemas.....	88
3. Terminar la Documentación del Sistema.....	88
4. Probar la Base de Datos en un Ambiente similar al Real.....	89
5. Entregar la Documentación al Usuario.....	90
Productos al Final de la Fase.....	90
<b>CAPITULO IV.....</b>	<b>91</b>
Técnicas de Apoyo.....	91
<b>PARTE I. ADMINISTRACION DE DATOS.....</b>	<b>92</b>
Planeación.....	93
Diseño Lógico.....	93
Certificación de la Calidad de los Datos.....	93
Capacitación.....	93
<b>PARTE II. NOMENCLATURA DE DATOS.....</b>	<b>94</b>
Técnica de Nombramiento de un dato.....	94
La Palabra Clase.....	95
La Palabra Entidad.....	95
Los Modificadores.....	95
<b>PARTE III. SEGURIDAD DE LOS DATOS.....</b>	<b>97</b>
<b>PARTE IV. ELKA.....</b>	<b>99</b>
Premisas para el desarrollo de ELKA.....	99
Cardinalidad 1:0:M.....	101

Cardinalidad I:M.....	102
PARTE V. DICCIONARIO DE DATOS.....	104
CONCLUSIONES.....	105
Bibliografía.....	106

## **CAPITULO I**

### **Conceptos Generales**

***... la aceptación de los límites personales es el primer paso hacia la madurez.***

Los conceptos generales por describir son :

De concepto y diseño, de estructura y almacenamiento, de procesamiento, de Explotación, y Plataformas.

### **PARTE I. DE CONCEPTO Y DISEÑO.**

#### **Entidad.**

Es todo objeto (real o abstracto, material o inmaterial) que produce y/o requiere información.

Por ejemplo: el empleado, una cuenta contable. Generalmente identificando una *entidad* se puede dar origen a una estructura de datos.

#### **Atributo de una Entidad.**

Es lo que califica a una entidad, por ejemplo la entidad EMPLEADO, tiene como *atributos* el nombre del empleado, el sexo, su número de nómina, etc.

#### **Ocurrencia.**

Conceptualmente es genérica, por ejemplo el EMPLEADO, el ALUMNO, la CUENTA CONTABLE, etc. Una *ocurrencia* indica la cuantificación de una entidad, por ejemplo hay una entidad empleado pero puede haber 100 empleados en una compañía, o también hay una entidad ALUMNO pero en la escuela puede haber 18,000 alumnos.

#### **Relación.**

Es la forma por la cual se usa y se comparte información entre dos más entidades. Ciertas entidades no pueden subsistir sin insumos (datos) o productos (información) de otras.

#### **Llave.**

Es un atributo de una entidad que tiene como función principal el que la ocurrencia sea única en una estructura donde se almacenan los datos.

Determinar un atributo como *llave* permite hacer uso de operaciones de *manipulación de información* como almacenamiento, búsqueda y recuperación en base a ella.

Por ejemplo, siguiendo con el tema de la escuela; podrá haber 18,000 alumnos pero solo hay uno con el número de cuenta 8323258-3.

El número de cuenta es la *llave* y es un atributo de la entidad alumno y podrá ser la ocurrencia 16680 pero únicamente lo identificamos por el 8323258-3. No es forzoso que sólo se determine una *llave* por entidad, se pueden determinar tantas *llaves* como la necesidad de información lo indique por ejemplo otra *llave* puede ser el atributo Nombre del Alumno.

#### ***Llave Primaria.***

Es uno de los atributos de la entidad que hace única a cada ocurrencia de la estructura de datos. Esta *llave* siempre es única. Un ejemplo de una *llave primaria* que podemos señalar es el Registro Federal de Contribuyentes que cada persona física debe tener y que es único.

#### ***Llave Alterna.***

Es uno de los atributos de la entidad que nos permite almacenar y recuperar a cada ocurrencia de la estructura de datos sin que esta defina la unicidad del registro, es decir, puede haber más de un registro con la misma *llave*.

Un ejemplo de *llave alterna* (o secundaria) puede ser dentro de una estructura de datos de alumnos el nombre del alumno o su dirección, tomando como *llave primaria* su número de cuenta.

#### ***Reglas de Integridad.***

Son todas aquellas reglas y normas que deben seguirse para la transformación de un dato en información. Un ejemplo de una *regla de integridad* contable es la de que en un asiento contable la suma de los cargos debe ser igual a la suma de los abonos.

Las *reglas de integridad* tienen como función darle validez a la información mediante el seguimiento de las reglas y normas que se definan en la empresa para el tratamiento de la información.

### **Reglas del Negocio.**

Son todos aquellos lineamientos que deben seguirse en el procesamiento de la información para el logro de los objetivos de una empresa.

Por ejemplo una *regla del negocio* de un sistema de nómina es que un empleado no puede trabajar mas de 10 horas extras en una quincena .

### **Modelo de Información.**

Es un diagrama que muestra la integración de entidades, atributos, llaves y relaciones, cuyo propósito es el de dar a conocer la composición de un negocio. Su objetivo es mostrar en forma clara y sencilla la información requerida en los procesos de la empresa.

No podemos forzar a representar un *modelo de información* de una manera fija, es necesario saber que cualquier técnica que se use para crear el modelo es válida, lo importante es representar de una forma abstracta el negocio.

### **Requerimiento .**

Un *requerimiento* en Ingeniería de Sistemas es la forma de nombrar a una necesidad de información. Esta necesidad puede estar representada de diferentes maneras: por *información, bienes materiales y servicios.*

Ejemplo de *requerimiento* de Información: un reporte, pantallas de consulta, gráficas, microfichas, archivos, listados, cintas, discos flexibles (conteniendo archivos).

Ejemplo de *requerimiento* de Bienes Materiales: autos, libros, calculadoras, computadoras etc.

Ejemplo de *requerimiento* de Servicios: el pago de luz en una sucursal bancaria, la alta de un alumno en la Licenciatura en Informática, compra y venta de acciones, reservación de un boleto de avión.

Incluso, un *requerimiento* puede ser la necesidad de un nuevo sistema o la modificación del actual.

**Software.**

Es una palabra del idioma inglés la cual no tiene traducción al español. Se usa dicha palabra para nombrar al conjunto de instrucciones que ejecuta una computadora. Dicho conjunto de instrucciones forma un programa.

Es muy importante hacer notar que en la actualidad las computadoras trabajan en base al *software* que tienen almacenado; dicho *software* es ejecutado por el procesador del computador para dar los resultados deseados.

Existen diferentes clases de *software* entre algunos ejemplos se encuentran: Sistemas Operativos, Software de Aplicación, Sistemas de Monitoreo, Administradores de Bases de Datos, Graficadores, Emuladores, Procesadores de Palabra, Hojas de Cálculo, Sistemas de Correo Electrónico, Software para Comunicaciones y Transferencia de Datos.

**Hardware.**

También es una palabra del idioma inglés. El uso que le dan en dicho idioma es para denominar a las herramientas de ferretería.

El *hardware* es una palabra de uso general para nombrar a todos los componentes físicos que integran a un sistema de computación.

Algunos ejemplos de *hardware* son: el monitor de la computadora personal, una impresora, un circuito integrado de silicio (CHIP), un ratón (mouse), una unidad de disco, el disco duro de una computadora personal, los cables de energía eléctrica que alimentan la unidad central de proceso.

**PARTE II. DE ESTRUCTURA Y ALMACENAMIENTO.****Bit.**

Es la unidad básica de información, cuyo valor incluye uno de dos dígitos posibles (0 y 1). Los dígitos "1" y "0" representan la condición de "presencia y no presencia" de energía eléctrica (de encendido y apagado).

También lo podemos relacionar a otros conceptos como el de que uno es igual a cierto y cero es igual a falso o si (1) no (0).

Al sistema que manipula los dos dígitos se le llama *Sistema Binario*. Como sabemos, nuestro sistema numérico tradicional es el *sistema decimal* porque usamos diez dígitos para representar nuestras cantidades (0,1,2,3,4,5,6,7,8,9).

El Sistema Binario es posicional ya que para hacer la conversión de binario a decimal se usa la posición del dígito binario .

Para dar una idea de los equivalentes *enteros no negativos* de binario y decimal podemos hacer el siguiente ejercicio .

En el Sistema Binario cada posición del *bit* representa una potencia de 2. El *bit* que está más hacia la derecha representa 2 a la 0, lo cual es igual a 1; el de la siguiente posición a la izquierda representa 2 a la 1, el cual es 2 y así sucesivamente. Un número entero es representado como una suma de potencias de 2 .

Por ejemplo el grupo de *bits* 00100110 tiene unos en las posiciones 1, 2 y 5 (contando de derecha a izquierda e interpretando la posición de más a la derecha como la posición 0) que representa el :

$$1*(2^{**1}) + 1*(2^{**2}) + 1*(2^{**5}) = 2 + 4 + 32 = 38.$$

No es necesario contar los ceros porque por definición todo número multiplicado por cero es igual a cero.<sup>2</sup>

Luego entonces podremos decir que la serie 00100110 en binario es el número 38 decimal.

---

<sup>2</sup> Utilizaremos la notación de  $x^{**}y$  para representar la exponenciación, es decir x elevado a la y .

Las computadoras pueden operar usando diferentes sistemas numéricos por ejemplo el binario (que contiene 0 y 1), octal (que contiene 0,1,2,3,4,5,6,7), hexadecimal (con los elementos 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) siendo los más comunes el binario y el hexadecimal.

Si deseáramos convertir de hexadecimal a decimal podemos utilizar el método anterior, siguiendo el mismo concepto posicional y usando potencias de 15.

Por ejemplo el número 0F1 hexadecimal sería representado por:

$$\text{Si } F = 15 \text{ posicionalmente, } 1 \cdot (16^{**}0) + 15 \cdot (16^{**}1) = 1 + 240 = 241.$$

Entonces el 0F1 hexadecimal es 241 decimal.

Existen dos estándares de códigos para el manejo de información en el ambiente de micro computadora y de mainframe, estos son el ASCII y el EBCDIC respectivamente.

**Ascii.**

Código con el propósito de estandarizar el intercambio de información entre los equipos producidos por diferentes fabricantes. Es un acrónimo del Estándar de Código Americano para el Intercambio de Información, [A(merican) S(tandard) C(ode for) I(nformation) I(nterchange) .] <sup>3</sup>

**Ebcdic.**

Código para la representación de información alfanumérica. Es un acrónimo de Código Extendido de Intercambio de Binario a Decimal, [e(xtended) + b(inary) + c(oded) + d(ecimal) + i(nterchange) + c(ode) .] <sup>4</sup>

Ambos códigos son los que más comúnmente se usan; en las computadoras personales se usa el *Ascii* y en los super computadores (mainframe) el *Ebcdic*.

Estos códigos mantienen un orden para el manejo de los caracteres, por ejemplo el caracter A .

---

<sup>3</sup> *The American Heritage Dictionary*, (Houghton Mifflin, 1987), Microsoft Bookshelf 1992, s.v. "ASCII."

<sup>4</sup> *The American Heritage Dictionary*, (Houghton Mifflin, 1987), Microsoft Bookshelf 1992, s.v. "EBCDIC."

Conociendo que el ordinal (el orden en el código Ascii) del caracter A es el número 65 podemos hacer el siguiente ejercicio.

Para *convertir de decimal a binario* podemos utilizar el siguiente método:

Haciendo divisiones sucesivas :

$65/2 = 32$  con un residuo de uno, el uno lo anotamos de derecha a izquierda

1

Así el cociente lo dividimos entre 2 y tenemos que

$32/2 = 16$  con un residuo de 0

01

Repetimos el procedimiento hasta que el cociente sea 1 y lo agregamos al final de la lista .

$16/2 = 8$  con residuo de 0

001

$8/2 = 4$  con residuo 0

0001

$4/2 = 2$  con residuo de 0

00001

$2/2 = 1$  con residuo de 0

000001

agregamos el 1 que nos sobró

1000001

Así sabemos que el caracter A con ordinal 65 decimal está representado en binario como 1000001.

Si deseamos convertir de binario a decimal para corroborar que todo esté bien hacemos lo siguiente:

$$1 * (2^{**0}) + 1 * (2^{**6}) = 1 + 64 = 65 .$$

### **Byte .**

Es el conjunto de ocho dígitos binarios o bits que dan origen a un caracter. Un byte tiene 256 combinaciones de "encendido-apagado". Cada byte puede representar uno de doscientos cincuenta y seis caracteres alfanuméricos.

Siempre se usan los ocho bits para representar el caracter, por ejemplo el 2 tiene el siguiente conjunto de bits 00000010.

### **Equivalencias .**

Es importante conocer las equivalencias de las cuantificaciones existentes en bytes.

KiloByte (KB), MegaByte (MB) y GigaByte (GB).

1 Byte=8 Bits= 1 Caracter ya sea alfabético ó numérico .

1 KB=1024 bytes=1024 Caracteres=1024 \* 8=8192 bits

1 MB=1024 Kbytes=1024\*1024=1,048,576 bytes=1,048,576 caracteres= 8,388,608 bits.

1 GB=1024 MB=1024 \* 1024 \* 1024=1,073,741,824 caracteres .

### **Caracter.**

Es la unidad mínima de información representada por una letra, número o símbolo. Por ejemplo la letra A, el número 9 o los símbolo \$, /, %, (, \*, etc. Así mismo existe otro caracter que es el espacio en blanco representado en la jerga informática con un b cruzada por una diagonal de derecha a izquierda.

**Palabra o Cadena.**

La palabra o cadena es el conjunto de uno o varios caracteres agrupados como una unidad. Por ejemplo la palabra "CASA", la cadena "PARANGARICUTIRIMICUARIO".

**Campo.**

Es el conjunto de uno o varios caracteres o bien el conjunto formado por una o varias cadenas. Por ejemplo el número 100 o la palabra "CASA", el enunciado "CERRADA DE LA ROMERIA" que podría ser el atributo de dirección de la entidad empleado. Así podemos tener el *campo* número de cuenta.

**Registro.**

Es el conjunto de uno o varios campos. Por ejemplo el *registro* de un alumno con los campos: nombre, número de cuenta, nombre de la escuela, etc.

**Dato.**

Un *dato* puede ser un caracter, campo o registro el cual no ha sido procesado. Cuando es procesado se convierte en *información*.

**Estructura de Datos .**

Es el área física dentro de la memoria ya sea principal o secundaria donde se almacena la información. Es la forma en la que se organiza la información dentro de la memoria de la computadora. Físicamente las *estructuras de datos* están relacionados dentro de la computadora por medio de los apuntadores.

**Información.**

Es el conjunto de uno o varios datos resultado de un procesamiento manual y/o automatizado.

**Archivo.**

C.J. Date menciona: "...un *archivo* es una colección de todas las ocurrencias de un tipo de registro almacenado...".<sup>5</sup>

---

<sup>5</sup> C.J. Date, An Introduction to Data Base Systems; Vol I; Fifth Ed; pag 59.

Es el conjunto de uno o varios registros en secuencia.

Anteriormente el uso de un *archivo* era un factor determinante en el procesamiento de un sistema de información, debido a que no se conocían otras estructuras de información. Actualmente los *archivos* son ampliamente usados para la transferencia de información entre diferentes plataformas de equipos de cómputo y entre diferentes bases de datos, convirtiendo algunas tablas de las bases de datos en "*archivos planos*"<sup>6</sup> y viceversa.

Como usuarios no debe importarnos donde y cómo se almacenará la información ya que ésta no es una de nuestras tareas. Como cultura general sólo debemos quedarnos con la idea que la información físicamente se almacena en *estructuras de datos*.

#### **Base de Datos .**

C.J. Date define al Sistema de *Base de Datos* "...es esencialmente no mas que un sistema computarizado de almacenamiento de registros. La base de datos en si misma puede definirse como una clase electrónica de archivado en gabinetes; en otras palabras, es un almacén para un conjunto de archivos de datos computarizados. El usuario del sistema tendrá las facilidades para realizar una variedad de operaciones con los archivos, incluyendo entre estas operaciones: insertar, recuperar, actualizar y borrar registros; así como agregar nuevos archivos y borrarlos de la base de datos..."<sup>7</sup> Así mismo posteriormente indica que "...el sistema de base de datos... cuyo propósito es mantener la información y hacer que este disponible de acuerdo a la demanda..."

Es el conjunto de registros interrelacionados por medio de campos llave; está diseñada para ser compartida y cumplir con las necesidades de diferentes usuarios.

Los datos están almacenados de tal manera que son independientes de las aplicaciones que los utilizan y deben estar agrupados en función de las características comunes de la empresa. Una *Base de Datos* está compuesta por diferentes elementos: Tabla, Índices y la Bitácora de Transacciones.

---

<sup>6</sup> Archivo Plano es un término de la jerga informática para indicar que es una serie de registros sin relación alguna.

<sup>7</sup> C.J. Date. An introduction to Database Systems; Vol I; Fifth Ed; Pag 3.

### COMPONENTES DE UNA BASE DE DATOS

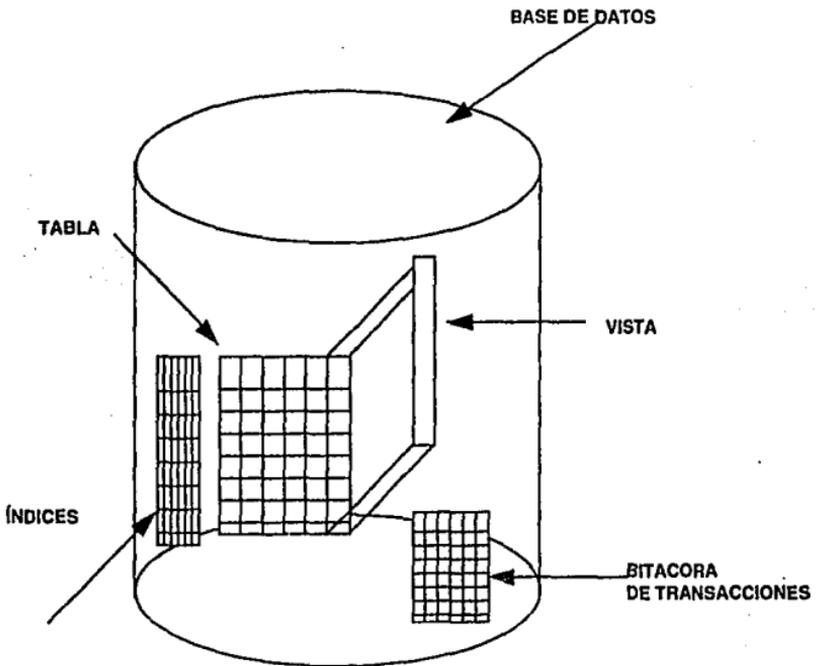


Figura 1. Componentes de una Bases de Datos.

#### Sistema Administrador de Base de Datos (DBMS),<sup>8</sup>

El DBMS (Data Base Managment System) es el sistema que controla la operación de la base de datos y los recursos de la misma, a su vez controla el acceso de los usuarios hacia los datos de la base.

<sup>8</sup> Idem. Pags 42 a 45.

Las funciones de la base de datos son las siguientes:

*Definición de Datos.*

*Manipulación de Datos.*

*Seguridad de los Datos.*

*Integridad de los Datos.*

*Recuperación de Datos.*

*Control de Concurrencia.*

*Diccionario de Datos.*

*Administración de los Recursos de Operación. (Performance)*

El DBMS incluye utilerías, herramientas de desarrollo de aplicaciones, ayudas para diseño, reporteadores, lenguaje de manipulación de datos, lenguaje de consulta, lenguaje de definición de datos.

#### **Tabla.**

La *tabla* es una estructura de datos donde se almacena la información dentro de la base de datos. Generalmente hay más de una *tabla* en una base de datos.

Dependiendo del análisis, una entidad por lo general da origen a una *tabla*, por ejemplo si determinamos la entidad empleado con sus diferentes atributos, podemos definir la *tabla* de empleados que contenga todas las ocurrencias (datos de los empleados) de la empresa.

En una *tabla* las ocurrencias se almacenan conforme van llegando a la estructura y se puede acceder a ellas por medio de los índices .

#### **Índices.**

Los *índices* sirven para almacenar, clasificar, buscar y recuperar los registros de una tabla. La estructura de *índices* contiene uno de los atributos de la entidad que es el campo llave y un campo que contiene la posición física (la dirección de memoria) de la ocurrencia en la tabla.

Pueden existir varios índices conforme tantas llaves se hayan definido en el análisis.

No es forzoso que una estructura tenga solo una llave, puede existir más de una.

**Vistas.**

Las *vistas* son estructuras de datos resultado del tratamiento de tablas de una base de datos. Son almacenadas en la memoria de la computadora y no son tablas en si mismas. El objetivo de las vistas en base de datos es el de crear una estructura lógica de información para facilitar la consulta del contenido de las tablas.

Por ejemplo podremos tener la base de datos de la empresa en donde haya tablas como: Vendedores, Productos y Clientes. Para saber cuanto ha vendido la empresa por cliente y por producto se debe hacer algunas operaciones en "lenguaje de consulta", ejecutando dicho proceso para producir la información. Sin embargo podría crearse una *vista* en el "lenguaje de consulta" para que permanentemente se tenga la información disponible para el Usuario.

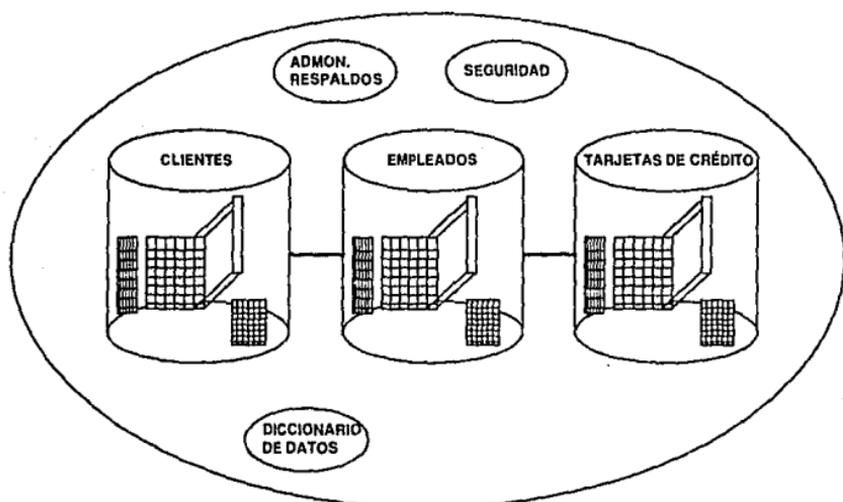
**ESQUEMA DE UN SISTEMA ADMINISTRADOR DE BASES DE DATOS**

Figura 2. Esquema de un sistema administrador de base de datos.

**Bases de Datos vs. Archivos.<sup>9</sup>**

¿Porqué usar Bases de Datos?, las bases de datos tienen algunas *ventajas* sobre los archivos entre algunas de ellas encontramos:

*Minimización del Espacio:* No se requiere de archivos voluminosos de papel.

*Velocidad:* La computadora puede recuperar y cambiar la información más rápido de lo que lo podría hacer un ser humano.

*Reducción del Trabajo:* Las tareas repetitivas de recuperación y almacenamiento de información pueden ser realizadas por los procesadores de las computadoras.

*Oportunidad:* Capacidad de obtener información actualizada, oportuna y disponible al momento en que se le requiere.

Así mismo existen algunos *beneficios* que se tienen al usar bases de datos:

*Reducción de Redundancia:* En sistemas con archivos cada uno tiene sus propios archivos privados, los cuales no pueden ser usados por todas las aplicaciones de una institución.

*Consistencia:* Es una consecuencia del beneficio anterior. Los sistemas de base de datos permiten dar un tratamiento determinado y fijo sobre la información que contienen, debido a que se procura que no haya duplicidad o redundancia de datos.

*Compartir Información:* Las aplicaciones existentes pueden compartir la misma información que reside en la base de datos. A su vez las aplicaciones futuras podrán diseñarse en base a la base de datos existente.

*Forzar al uso de Estándares:* Creando una base de datos central, se puede asegurar que se aplicarán todos los estándares institucionales.<sup>10</sup>

*Seguridad:* Teniendo un control completo sobre la base de datos se puede asegurar que (a) se accese a los datos exclusivamente por los medios permitidos para ello y (b) que se controle la información que el personal accesa.

---

<sup>9</sup> Idem. Pags. 9 a 17.

<sup>10</sup> Si existieran en la organización, algunos ejemplos de estándares pueden ser: los de documentación, nombres de variables, longitud de campos, etc.

**Integridad:** La integridad es asegurar que los datos que se encuentran en la base de datos es segura, es decir, que el dato que se encuentra almacenado es el que se ha planeado que resida ahí; para lograr lo anterior es necesario que se realicen chequeos de integridad, es decir validaciones de la información.

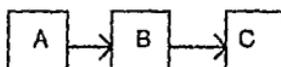
#### **Apuntador.**

Técnicamente un *apuntador* es una dirección de memoria que nos dice la posición física de la información en la memoria del computador. Como usuarios debemos conceptualizar a un *apuntador* como un objeto que nos lleva o nos conduce a otro.

Otra función de los *apuntadores* es la de representar la relación entre dos o más entidades.

#### **Listas.**

Es una colección de elementos, uno detrás de otro. El orden de los elementos de una *lista* está referenciado por los *apuntadores*, así podemos tener el siguiente ejemplo .



**Figura 3. Ejemplo de una Lista.**

El elemento A nos conduce al B y él a su vez nos conduce a C. Si yo deseara llegar al último elemento de la *lista*, sería necesario recorrer todos los elementos de la misma.

Otro tipo de *listas* son las *doblemente ligadas*, estas *listas* tienen *apuntadores* en ambos sentidos .



**Figura 4. Ejemplo de una Lista Doblemente Ligada.**

**Arreglos.<sup>11</sup>**

Los *arreglos* son un tipo de lista que reúne elementos *de la misma naturaleza*, son también conocidos como vectores, por ejemplo un *arreglo* de números, un *arreglo* de letras (que da origen a una palabra).

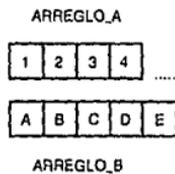


Figura 5. Ejemplo de un Arreglo.

Un ejemplo de los *arreglos* puede ser una palabra, en la jerga de sistemas también se le conoce como *cadena* que es una *lista* ligada de caracteres.

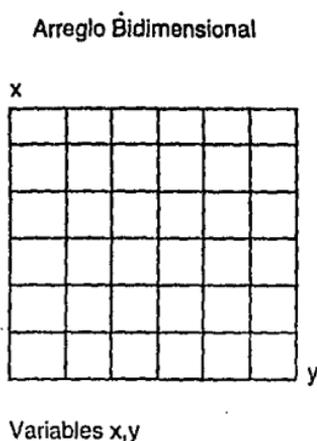
Los *arreglos* pueden ser uni-dimensionales (de una dimensión) como lo es el vector de la figura anterior. Dependiendo del número de vectores que se interrelacionen los *arreglos* pueden ser *arreglos* bi-dimensionales (de dos dimensiones), tridimensionales y n-dimensionales (de n dimensiones).

Los *vectores* son representados por las *variables* del plano cartesiano. A éstas *variables* se les conoce como *dimensiones*.

Un ejemplo de arreglos bi-dimensionales y tridimensionales es el siguiente.

---

<sup>11</sup> El concepto de Arreglo va mucho mas allá de lo que muestro, lo importante es comprender la interrelación de variables de una manera abstracta .



*Figura 6. Ejemplo de un Arreglo Bidimensional.*

Los arreglos bi-dimensionales también son conocidos como **MATRICES**.

Ahora que vemos como es un arreglo bi-dimensional de una manera abstracta, imaginemos una aplicación de este concepto. Un ejemplo puede aplicarse a los recursos humanos usando dos diferentes variables, la variable Y puede ser los empleados de la empresa, la variable X pueden ser los puestos que ocupan los empleados en la organización.

En el ejemplo, si deseamos saber cuál es el puesto de algún empleado solo basta con hacer la intersección entre la variable X y Y, tal intersección está representada por el círculo relleno con líneas diagonales.

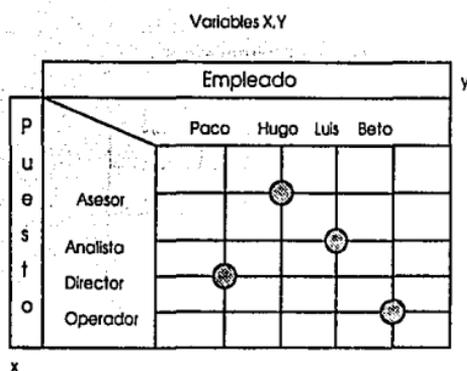


Figura 7. Ejemplo de la Aplicación de un Arreglo Bidimensional.

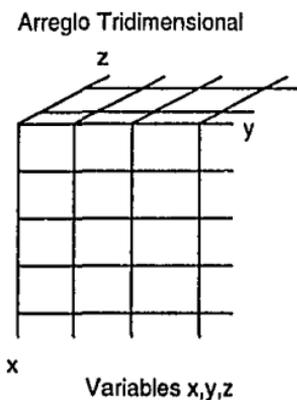


Figura 8. Ejemplo de un Arreglo Tridimensional.

Para ejemplificar la aplicación de un arreglo tridimensional podemos ver lo siguiente:

Imaginemos que nos encontramos inmersos dentro de la contabilidad de una institución bancaria y deseamos interrelacionar tres variables para representar los movimientos contables de la entidad: moneda, cuenta y sucursal.

## Ejemplo de un Arreglo Tridimensional

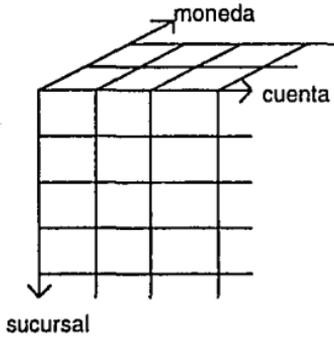


Figura 9. Ejemplo de la Aplicación de un Arreglo Tridimensional.

**Pilas.**

El concepto de una *pila* es tan simple como el de estibar varias cajas.

Las *pilas* son un tipo especial de listas que son una serie de elementos que se van agrupando conforme van llegando; el primer elemento en llegar es el último en salir.

Para usar estas estructuras hay dos métodos: El de meter y sacar (push y pop), en éste caso para llegar al primer elemento que arriba a la estructura es necesario sacar todos los elementos que le preceden.

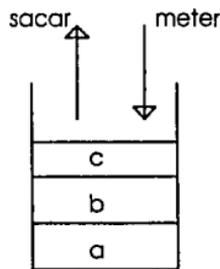


Figura 10. Ejemplo de una Pila.

Las pilas se usan para la optimización del espacio en disco y para el manejo de la memoria RAM de la computadora .

### Colas.

Las *colas* también son muy simples, ya que basta con imaginar cuando hacemos fila en espera de entrar a un estacionamiento, el primero en llegar es el primero en ser atendido.

Las *colas* tienen la característica de que el primero en llegar es el primero en salir. Hay dos tipos de *colas*, las de una entrada y las de doble entrada.

Un ejemplo de su uso: Las *colas* son muy importantes en el uso de una impresora compartida, existen métodos para "encolar" los requerimientos de impresión conforme van llegando.

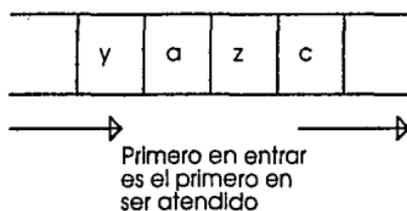


Figura 11. Ejemplo de una Cola.

### Una Breve Historia de la Computación.<sup>12</sup>

Aunque el desarrollo de las *computadoras* digitales se basa en el ÁBACO y en viejas calculadoras mecánicas, Charles BABBAGE fue quien diseñó la primer *computadora* moderna. La primer calculadora automática completa fue la MARK I, o la Calculadora Automática Secuencial creada en 1939 en Harvard por Howard Aiken, mientras que en 1946 se terminó la primer *computadora* electrónica digital de propósito general en la Universidad de Pennsylvania dándole el nombre de ENIAC (Calculador Integrador Numérico Electrónico) que contenía cientos de *tubos de vacío*. A las *computadoras* creadas en base a los tubos de vacío se les conoció como las de la *primer generación*.

<sup>12</sup> La historia se sucedió en los Estados Unidos de Norteamérica.

Posteriormente a principios de 1951 se creó la primer *computadora* que manipuló datos numéricos y alfabéticos con igual facilidad.

La *primer generación* de computadoras fue sustituida por las *computadoras de transistores* a finales de los 50s y principios de los 60s. A las computadoras de transistores se les llamó máquinas de *segunda generación* las cuales podían ejecutar millones de instrucciones por segundos. Las máquinas de la *tercera generación* remplazaron a las de la segunda a mediados de los 60s y principios de los 70s. Las computadoras de la tercera generación contienen *circuitos integrados*.

Los 80s se caracterizaron por el desarrollo y auge del microprocesador, las minicomputadoras y de la computadora personal.

### *Computadora.*<sup>13</sup>

Es un dispositivo capaz de ejecutar una serie de operaciones lógicas y aritméticas; se distingue de una calculadora debido a que tiene la posibilidad de almacenar los programas y a que puede almacenar y recuperar datos sin la intervención humana, permitiéndole ejecutar operaciones repetitivas haciendo uso de la lógica que se le indica.

Los cuatro principales componentes físicos de la *computadora* son el *CPU*, la *memoria principal*, la *memoria secundaria* y los *dispositivos de entrada y salida*.

Las operaciones se ejecutan en la Unidad Central de Proceso (CPU), el cual contiene circuitos lógicos que realizan las operaciones aritmético lógicas y para el control de los periféricos del sistema de computación.

La Unidad Central de Proceso también contiene *registros*, que son lugares donde se almacena la información que puede recuperarse de una forma más rápida que la RAM y son usados para retener los resultados intermedios de los cálculos.

El almacenamiento principal se encuentra en la unidad de almacenamiento o *memoria*. La memoria se almacena en pequeños *circuitos integrados* (CHIPS), los cuales contienen cientos de *semiconductores* que representan un bit.

---

<sup>13</sup> *The Concise Columbia Encyclopedia*, (Columbia University Press, 1991), Microsoft Bookshelf 1992, s.v. "computer."

Los programas y los datos que están almacenados en la memoria principal y que no son usados en el momento son guardados en la memoria principal o usando dispositivos de almacenamiento externo.

Los datos son introducidos y procesados dentro de la *computadora* vía los *dispositivos de entrada y salida*.

Los programas de *computadora* (software) controlan el funcionamiento del hardware y dirigen su operación.

#### ***BootStrap. (Autoinicialización)***

Es el proceso de encendido de una computadora. El CPU tiene un chip ROM que tiene grabado el auto reconocimiento de los dispositivos conectados al CPU. Se conoce como BOOT, o en español como BOOTear.

#### ***Memoria.***

Los datos se transmiten de diferentes maneras al CPU (Unidad Central de Proceso), al llegar a él no todos son usados de manera inmediata, parte de estos datos deben ser almacenados para su uso posterior. El lugar donde se almacena la información es denominado *Memoria*; la *memoria* se clasifica en *Memoria Principal* y *Memoria Secundaria*.

Los programas que están siendo ejecutados por el CPU, así como los datos necesarios para él deben residir en la *memoria* principal. La información que no se requiere de inmediato se delega a los dispositivos de almacenamiento externo (o secundario) donde el tiempo de recuperación y respuesta es un poco mayor.

Algunos otros dispositivos de almacenamiento secundario son : *Discos Duros, Discos Flexibles, Cintas Magnéticas, Cartuchos Magnéticos, Disco Compacto, Discos Ópticos*.

#### ***RAM.***

Este es un acrónimo de las palabras RANDOM ACCESS MEMORY, sencillamente significa la *memoria* de acceso aleatorio. Sin entrar en más detalles la RAM es la *memoria* de acceso rápido donde se almacena la información. Esta *memoria* es volátil, es decir, está activa siempre y cuando esté prendida la computadora; si se apaga la máquina está *memoria* se pierde. En la *memoria* RAM se carga el sistema operativo de la computadora.

**ROM.**

Es el acrónimo de las palabras READ ONLY MEMORY, es la memoria de sólo lectura, a éste grupo pertenecen algunos CHIPS que permiten que la computadora reconozca y habilite sus periféricos (proceso llamado de Autoinicialización) y los Disco Compacto.

A estos dispositivos sólo se les graba información una vez y posteriormente siempre servir sólo de lectura.

**Disco Duro .**

Es un periférico donde se almacena la información de manera permanente, es diferente a la RAM, ya que esta memoria no es volátil.

**Disco Flexible.**

También llamado diskette o floppy. Este es un dispositivo de almacenamiento secundario de menor capacidad que un disco duro; es removible de los drivers de la computadora. Los *diskettes* pueden ser de diferentes tipos y combinaciones :

**Formatos** : 5 1/4 y 3 1/2 pulgadas .

**Densidad** : Baja, Alta.

**Lados** : Sencillo, Doble

1 de 5 1/4 de Baja densidad de doble lado tiene la capacidad de almacenar 360 KB o sea  $360 * 1024 = 368,640$  caracteres .

1 de 3 1/2 de baja densidad de doble lado tiene la capacidad de almacenar 720 KB o sea  $720 * 1024 = 737,280$  caracteres.

1 de 5 1/4 de alta densidad de doble cara tiene la capacidad de almacenar 1220 KB (1.2 MB) o sea  $1220 * 1024 = 1,249,280$  caracteres .

1 de 3 1/2 de alta densidad de doble cara tiene la capacidad de almacenar 1440 KB (1.4 MB) o sea  $1440 * 1024 = 1,474,560$  caracteres .

### Sistema Operativo.

Es el sistema administrador de los recursos de la computadora, esto incluye el CPU, monitor, teclado, impresora y demás periféricos conectados al computador. Es el programa que se encarga de la asignación de los recursos que reconoció el CPU al encenderse (Autoinicialización).

### Flujo General de Operación de una Computadora.<sup>14</sup>

Las tres operaciones básicas que componen el flujo de operación de una computadora son la *entrada*, *procesamiento* y *salida*. El término entrada (input) define la introducción de datos a la computadora. El procesamiento se refiere al tratamiento que se le da a los datos dentro de la unidad central de proceso (UCP/CPU). La salida se refiere a la información resultado del procesamiento en el CPU.

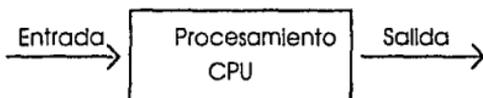


Figura 12. Flujo General de Operación.

Los medios por los cuales ingresa o egresa información del CPU son conocidos como *dispositivos*.

Los dispositivos de entrada tienen como función hacer la traducción de la información introducida a un formato entendible para el CPU (formato binario); para que una vez en formato binario el CPU lo procese. Así mismo los dispositivos de salida tienen como función la traducción del formato binario a un formato entendible para el usuario.

Existen diferentes dispositivos de entrada y de salida, así como clases de CPU<sup>15</sup>. La siguiente tabla muestra algunos de ellos.

<sup>14</sup> El flujo que presento es un concepto muy básico.

<sup>15</sup> Las clases pertenecen a la tecnología Intel de Computadoras Personales.

Clases de CPU	Dispositivos de Entrada	Dispositivos de Salida
8088	Teclado	Monitor o Pantalla
80286	Raton (Mouse)	Impresoras
80386	Discos Duros y Flexibles	Discos Duros y Flexibles
80484	Disco Compacto	Bocinas
80586 ó Pentium	Micrófono	Unidades de Cinta
	Scanner	Terminal
	Terminal	
	Unidades de Cinta	

Tabla 1. CPU's, Dispositivos de Entrada y de Salida.

#### La Unidad Central de Proceso (UCP/CPU).<sup>16</sup>

Todas las computadoras cuentan con una unidad central de proceso, es ahí donde se realiza el tratamiento a la información. El CPU está compuesto por tres componentes: la *unidad de control* (UC), la *unidad aritmética lógica* (UAL) y la *unidad de almacenamiento primario* (UAP).

Orilia menciona que "...la UC gobierna todas las actividades de la computadora y supervisa la ejecución de los programas. Controla y controla al sistema de cómputo de la misma manera como el cerebro controla al cuerpo, ... primero, determina la operación que se debe realizar y a continuación obtiene la información necesaria para ejecutar la instrucción. Determina donde se guarda la información y la transfiere desde esa localidad. También se asegura de que los datos se transfieran correctamente de una localidad de almacenamiento a otra. Una vez ejecutada la instrucción, la UC debe guardar los resultados para efectuar la salida o para utilizarlos después. Por último debe localizar la siguiente instrucción a procesarse. La UC repite esta serie de operaciones para cada instrucción."

La UC coordina directamente las actividades de entrada y salida de un sistema de cómputo.

Orilia comenta que "...la ALU lleva a cabo todas las operaciones aritmético y lógicas. Las *operaciones aritméticas* que ejecuta incluyen la adición, sustracción, multiplicación, división y exponenciación. Una *operación lógica* es una comparación de dos valores, es un intento de determinar la relación entre dos valores..."

<sup>16</sup> Orilia, Las computadoras y la Información; 3a. Ed; Cap 3 pag. 101.

"..la UAP es el lugar donde se almacena la información que va a ser utilizada por la ALU y por la UC, ya que estas últimas no la almacenan. La UAP recibe y almacena los datos de entrada antes de procesarlos. También conserva los resultados del procesamiento antes de que pasen al dispositivo de salida y también se almacenan los resultados intermedios para su posterior utilización. Cualquier programa que esté siendo procesado debe permanecer en almacenamiento primario durante su ejecución..."

**PARTE III. DE PROCESAMIENTO.****Proceso.**

Un *proceso* es la realización o ejecución de cualquier solicitud hecha al procesador del Computador. Recordemos que la única forma de hacer solicitudes al procesador es mediante el software.

**Mensaje.**

Un *mensaje* es cualquier tipo de información el cual fluye a través de un medio de comunicación. Para que exista un *mensaje* debe de haber un emisor y un receptor. El emisor o receptor puede ser cualquier entidad de software y hardware. Los procesos que se realizan en el procesador se comunican mediante *mensajes*.

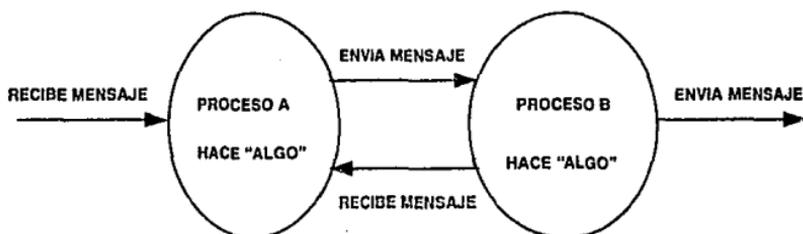
**EJEMPLO DE DIÁLOGO ENTRE PROCESOS**

Figura 13. Ejemplo de diálogo entre procesos.

**Programa.**

Son el medio por el cual se le indica al procesador y a sus periféricos la serie de instrucciones que deseamos que realice ante determinado evento. Los *programas* también son conocidos como el SOFTWARE.

Hay 3 diferentes estados por los cuales puede pasar un *programa* : Programa Fuente, Programa Objeto, Programa Ejecutable.

**Programa Fuente.**

Es el código que es grabado dentro de cualquier computadora usando un editor. Este código es comprensible para los humanos ya que generalmente está en lenguaje natural (español, inglés, etc.).

**Compilador.**

Cuando nos hablen de un *compilador* debemos pensar en un sistema que recibe como insumo un programa fuente al cual le da un tratamiento especial (lo compila) y obtiene un producto llamado programa objeto y/o ejecutable. Los *compiladores* revisan la sintaxis (que las instrucciones estén bien escritas) y posteriormente la lógica de las instrucciones.

*Los compiladores no revisan la lógica del programa, sino más bien la lógica de las instrucciones; muchas veces la lógica de las instrucciones no es la correcta aunque si lo sea la lógica del programa, o viceversa.*

Una vez que el *compilador* revisa la sintaxis y la lógica, se genera el programa objeto y algunos compiladores automáticamente generan el programa ejecutable. *Los compiladores revisan primero el programa fuente y generan el ejecutable, pero no ejecutan las instrucciones.*

**Programa Objeto.**

Es la segunda etapa por la cual pasa un programa. El programa fuente debe ser compilado y convertido a lenguaje de máquina, aunque este no es la última etapa por la cual debe pasar un programa ya que debe ser convertido código ejecutable .

**Proceso de Ligado.**

También conocido como proceso de "linkeo", dicho proceso de ligado consiste en convertir el código del programa objeto en código o programa ejecutable. En dicho proceso, si fue previamente especificado en el programa fuente, se asocian e interrelacionan las funciones contenidas en librerías y se agregan en el código ejecutable.

### **Librerías.**

Son rutinas que no son ejecutables, que contienen código de funciones y procedimientos que realizan algunas tareas; las *librerías* son llamadas desde los programas fuente, utilizando parámetros como medio de comunicación entre los programas fuente y las *librerías*.

### **Programa Ejecutable.**

Es la última etapa por la cual pasa un programa. El *programa ejecutable* es el que indica que instrucciones debe realizar el sistema operativo, previamente indicadas en el programa fuente.

Los programas fuente pueden ser traducidos usando dos tipos de herramientas, los compiladores y los intérpretes.

### **Intérprete.**

Un *intérprete* como su nombre lo indica *interpreta, traduce e inmediatamente ejecuta la instrucción*. El *intérprete* no genera programa objeto ni programa ejecutable. Tiene la desventaja ante el compilador de que si hay un error este se detecta al momento de la ejecución.

Actualmente es posible crear programas ejecutables para realizar una determinada función; sonaría ilógico siempre volver a crear un programa para hacer una tarea que en repetidos sistemas se requiere.

Para dar respuesta a esta situación existe otro tipo de programas: *las Librerías*.

### **Transacción.**

Una *transacción* es una operación generada por una entidad dentro de un sistema de información. La *transacción* siempre afectará a uno o varios registros de la base de datos.

Como ejemplos de una *transacción* podemos tener:

- A un asiento contable, en donde hay una interrelación de entidades como la Cuenta Contable, el Contador y el Diario Mayor por ejemplo .

- Un pago de Cheques dentro de una Institución Bancaria es una transacción que genera una serie de asientos contables y registro de operaciones en la aplicación de Cheques.

- Un alta de un empleado a nuestra base de datos de nómina.

Es muy importante tener presente el concepto de una *transacción* ya que actualmente todos los procesos en línea y tiempo real están basados en transacciones.

#### **Bitácora de Transacciones.**

Es el lugar físico donde se almacena la historia de las transacciones que actualizaron a una base de datos. Esta estructura de datos tiene varios usos:

- Guardar una historia de los movimientos que afectaron a la base de datos.
- Servir como soporte para que en caso de pérdida de los datos o falla del sistema, se pueda efectuar un rollback. Un *rollback*<sup>17</sup> tiene dos finalidades, el primero es el de regresar la base de datos a un estado anterior y el otro recuperar la base de datos y las transacciones que se efectuaron hasta el momento de la falla.

**Tipos de Procesamiento:** Proceso Local, Remoto, Batch, en Línea, en Tiempo Real y Distribuido .

#### **Proceso Local.**

El proceso local es aquel que se realiza en el procesador donde se generó la solicitud. El proceso local no requiere líneas de comunicación externa hacia otros procesadores para realizarse.

---

<sup>17</sup> Dicha palabra no tiene una traducción literal del inglés al español, podremos darle el significado de "ir de regreso e iniciar".

## EJEMPLO DE UN PROCESO LOCAL

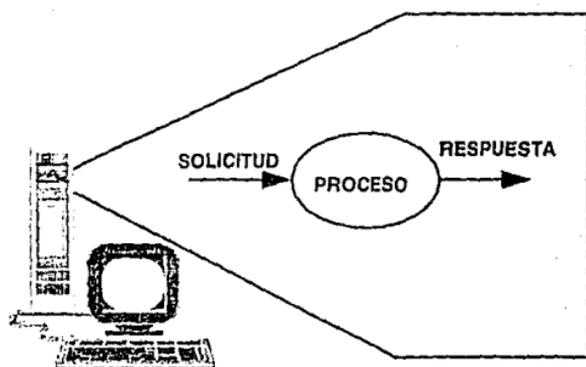


Figura 14. Ejemplo de un proceso local.

**Proceso Remoto.**

Un *proceso remoto* tiene mucho que ver con el esquema de comunicaciones entre Host<sup>18</sup>. Este tipo de procesamiento ocurre cuando dos o más host están conectados entre sí<sup>19</sup>, y alguna persona lanza un proceso desde un Host al cual puede llegar solo conectándose por el otro. La diferencia entre el proceso distribuido y el *proceso remoto* reside en que éste último toma el control de la transacción sin la necesidad de que el proceso solicitante requiera de respuesta del proceso ejecutante; el proceso distribuido sí requiere de la respuesta del proceso ejecutante al solicitante.

Supongamos que alguien se encuentra en Guadalajara donde existe un Host al cual tiene una terminal (o estación de trabajo) conectada, así otro Host se encuentra en el D.F., y el usuario decide lanzar un proceso del Host del D.F., desde la terminal de Guadalajara.

**Proceso Batch .**

Es el tipo de procesamiento en el cual se ejecutan una o más instrucciones en secuencia, una detrás de la otra. Podríamos dar un ejemplo de un *proceso batch* en el área de Recursos Humanos:

<sup>18</sup> Recuérdese que el Host, en este caso, puede ser un Server de LAN o un Mainframe.

<sup>19</sup> Formando una red de Host.

- Durante una quincena en alguna estructura de datos almacenamos la información de la actividad de nuestros empleados como altas, bajas y cambios, incapacidades, horas extras, días trabajados, etc. Así al finalizar la quincena debemos obtener nuestra nómina quincenal y pagar impuestos. Luego entonces podemos ejecutar nuestro *proceso batch* en el cual se actualizaran todas nuestras estructuras y tener nuestra nomina por empleado .

- En la contabilidad podemos dar un ejemplo de *proceso batch*. La SHCP pide como una obligación a una Institución Bancaria, que sus estados financieros se publiquen en determinada fecha. Aunque el sistema de contabilidad este en línea, lógicamente debe haber un proceso de cierre y de inicio de operaciones del mes. El ejemplo del *proceso en batch* haría lo siguiente: Dejar en ceros todas las cuentas contables del mes anterior e iniciar los saldos del mes con los del mes anterior .

- Otro ejemplo de un *proceso batch* podría ser el de un respaldo de una base de datos, donde la base de datos tiene diferentes estructuras y se crea un batch para indicar que estructuras debe de respaldar a cinta .

#### **Proceso en Línea .**

Un *proceso en línea* tiene como principal característica que actualiza las transacciones en las estructuras de datos al momento en que se le indica al computador, dicho momento depende del tipo de aplicación y del cómo fue diseñada. El *proceso en línea* puede hacer uso también del proceso batch, es decir, puede hacer uso de estructuras temporales para almacenar información y posteriormente ejecutar un proceso que actualice la información usando el contenido de la estructura temporal.

Así mismo el *proceso en línea* puede actualizar la información al momento en que se realice la transacción. El *proceso en línea* usa la bitácora de transacciones y la base de datos para mantener la integridad de la información.

#### **Proceso en Tiempo Real.**

Es un proceso en línea con la condición de que el acceso a la línea esta definido para un ambiente multiusuario y de multiprocesamiento. Esto es, hay más de un usuario accedando la base de datos y además hay más de un proceso corriendo en el computador.

El *proceso en tiempo real* excluye totalmente el proceso en batch, y tiene como condición especial que la información contenida en las transacciones es actualizada al momento en que se realiza la transacción.

**Proceso Distribuido.**

Para C.J. Date el *proceso distribuido* "...indica que se pueden interconectar muchas máquinas dentro de un red para ejecutar entre ellas una simple tarea de procesamiento de datos...".<sup>20</sup>

Para que exista por si el *proceso distribuido* deben existir dos tipos de proceso: *El Solicitante y el Ejecutante*. El solicitante es el que requiere la actuación de otro proceso para recibir cualquier tipo de mensaje. El Ejecutante es aquel que recibe el mensaje del solicitante para realizar cualquier tarea, con la condición de que envía un mensaje de regreso.

Un ejemplo de éste tipo de proceso puede ser el siguiente:

El Cobro de un Cheque.- Digamos que "Hugo" extiende un cheque a favor de "Paco" en el D.F., el cual entrega el cheque a "Luis" quien lo trata de cobrar en Hermosillo Sonora, (suponiendo que el "Banco" tiene la información de sus clientes en el Host A del D.F.); para realizar el cobro del cheque, el cajero de la sucursal de Hermosillo hace una solicitud del saldo de la cuenta de "Hugo" para pagarle a "Luis". Dicha solicitud implica que el Host B de la sucursal envíe un mensaje hacia el Host A del D.F., la transacción se ejecuta en el D.F., así se le da el efectivo a "Hugo", luego entonces el sistema del "Banco" hace una transacción al sistema contable que reside en el Host C del D.F.

---

<sup>20</sup> C.J. Date, An Introduction to Database Systems; Vol I; Fifth Ed.; pag 48.

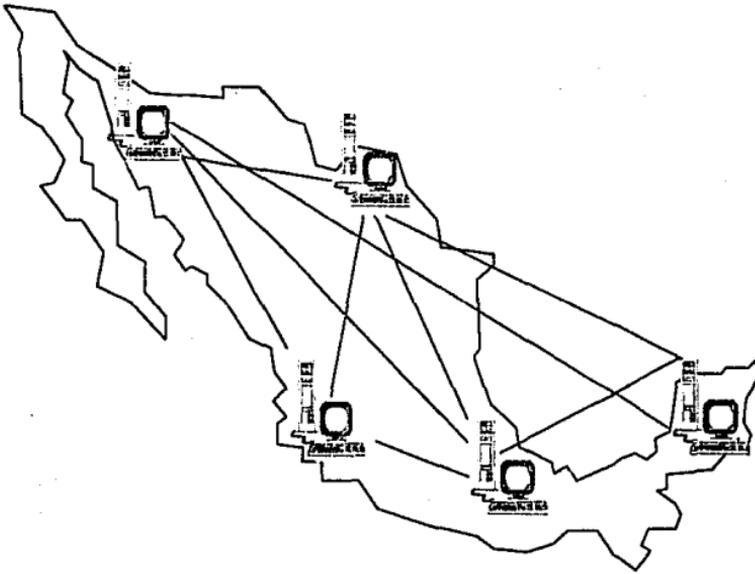


Figura 15. Ejemplo del proceso distribuido.

**PARTE IV. PLATAFORMAS.****Las Micro Computadoras o Computadoras Personales.**

Actualmente una computadora personal puede operar en diferentes formas y plataformas: Como PC Mono Usuario, Servidor de Archivos, Estación de Trabajo y Terminal Inteligente.

**Como una PC Stand Alone (Mono Usuario).**

Es una computadora personal que trabaja únicamente con los recursos que trae de fábrica. Únicamente puede acceder los archivos y aplicaciones que le permitan su arquitectura, esto las hace un poco limitadas.



Figura 16. Ejemplo de una PC mono-usuario.

**Como un Servidor de Archivos .**

Es una computadora personal que con la ayuda de un sistema operativo multiusuario, bajo la plataforma de un sistema operativo mono usuario y con el hardware necesario, presta servicios de compartición de recursos. En el servidor se almacenan las aplicaciones y los datos que se desean compartir.

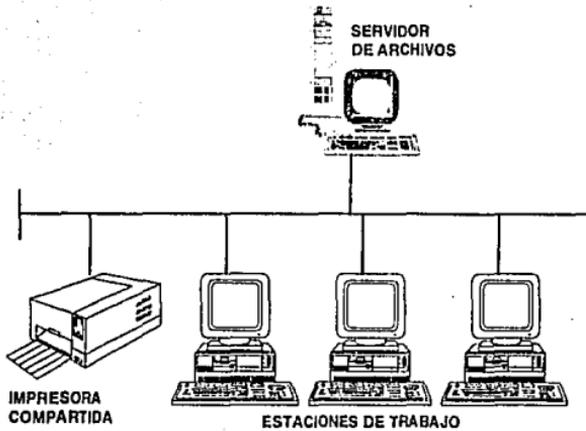


Figura 17. Ejemplo de un Servidor de Archivos.

#### Como una Estación de Trabajo.

Es una computadora personal que además de trabajar con sus propios recursos, usa los de un servidor de archivos. También se le conoce como NODO.

Haciendo un pequeño paréntesis acerca de un Servidor y las estaciones de trabajo; el Servidor no debe estar super equipado para que el tiempo de respuesta de la Red sea óptimo; son las estaciones de trabajo las que deben de estar bien equipadas, con que el servidor esté medianamente equipado es suficiente. Esto se debe a que el Sistema Operativo para compartir los archivos, solo se encarga de enviar copias a las estaciones de trabajo y son ellas las que procesan la información de una manera local .

#### Como una Terminal Inteligente .

Es una computadora personal ya sea Mono Usuario o Estación de Trabajo que usa un emulador de Terminal para conectarse con un Mainframe. Utilizan sus recursos o los del servidor para hacer más eficiente la comunicación con el Mainframe.

Algunos programas además de permitir la emulación de terminal, tienen la capacidad de poder transferir archivos de la computadora personal al Mainframe y viceversa. En algunas empresas las terminales tontas tienden a ser desplazadas por las terminales inteligentes.

**Mainframe.**

Es un computador de gran escala. Hay diferentes tipos y marcas de *mainframe*, también se lo conoce como HOST (computadores anfitriones). Este computador también tiene conectado dispositivos periféricos como procesadores, dispositivos de comunicación, discos externos, unidades de cinta, impresoras y terminales.

**Host.**

Un *Host* como su palabra lo indica es un Anfitrión, el concepto de Host esta íntimamente relacionado como símil a un Mainframe, aunque en diversos esquemas de Redes de Comunicación se toma como *Host* al computador central de la Institución o Empresa.

**Terminal.**

Conocidas como terminales tontas. Están compuestas generalmente por dos elementos: el monitor (pantalla) y el teclado. La mayor parte de las *terminales* no cuentan con un microprocesador ni con memoria para trabajar, únicamente operan como interface entre el humano y el mainframe.

**Red de Area Local.**

(LAN) Significa Local Area Network. Es una serie de estaciones de trabajo conectadas a un Servidor. En una red local se comparten archivos, aplicaciones y los recursos del Servidor como pueden ser dispositivos de comunicación, impresoras, correo electrónico y bases de datos. La característica que la hace de área local es que generalmente este tipo de redes es de pequeño o mediano alcance instalados en diferentes pisos de un edificio.

**Red de Redes.**

(WAN) Significa Wide Area Network o Red de Gran Alcance . Es una serie de Redes locales conectadas entre si, permitiendo que los usuarios de un servidor puedan acceder y usar los recursos de otro Servidor. Las WAN conectan varios servidores a mayores distancias.

## **CAPITULO II**

### **Los Sistemas de Información**

***.... actualmente quien tiene la Información tiene PODER.***

---

## PARTE I . COMPONENTES DE UN SISTEMA DE INFORMACIÓN.

Para poder entender el concepto de *Usuario e Ingeniero de Sistemas*, es necesario que definamos lo que es un *sistema*.

### *El Sistema.*

Es la interrelación de los objetos que existen en el Medio Ambiente por la cual los objetos logran un objetivo específico<sup>21</sup>. Este concepto lo considero el más apropiado para generalizar todos los esquemas posibles de un *sistema*.

Por Ejemplo:

- Si determinamos como *objetos* a las unidades organizacionales de una empresa y a esta última definirle un *medio ambiente* que podría ser su relación con Proveedores, Bancos y la Secretaría de Hacienda y Crédito Público.
- También es posible aplicarlo al sistema del cuerpo humano, el cual tiene *objetos* relacionados como son el aparato respiratorio, digestivo, auditivo y los sentidos del gusto, tacto, vista, oído y olfato. Determinándole al ecosistema como el *medio ambiente* con el cual tiene interacción.

Existen tres condiciones fundamentales para que un *sistema* opere y que determinan su comportamiento en el medio ambiente, estos son:

- *Insumo* (Entradas).
- *Proceso*.
- *Producto* (Salidas).

### *El Insumo.*

Es la *materia prima* que sirve como entrada de un sistema; la interrelación de los objetos que interactúan dentro de un sistema está determinada por la forma en la que un proceso recibe un *insumo* para producir algo.

### *El proceso.*

En el medio ambiente los objetos se interrelacionan y un sistema requiere de insumos para subsistir, pero ¿cómo y por qué puede un sistema mantenerse activo?

---

<sup>21</sup> Nota del Autor: He llegado a este concepto en base a la experiencia.

Los objetos por lo general tienen asignada una tarea específica dentro de la operación de un sistema, existiendo algo dentro de él que los clasifique y les de un tratamiento especial. Dicho *tratamiento* es fundamental para generar un producto que a la vez puede servir como insumo de otro proceso y continuar así la vida del sistema.

#### **El producto .**

Podríamos concluir que la razón de ser de un sistema es *PRODUCIR*. Todos los sistemas tienen como objetivo operar un insumo y dejar un *producto*, de ahí su origen .

Aplicando el concepto anterior a la contabilidad, por ejemplo, para que se realice un asiento contable se requiere de uno o varios cargos y/o varios abonos; en este caso el insumo es el cargo y el abono, siendo el asiento contable el producto procesado por una regla contable.

Otro ejemplo puede presentarse en los sistemas como el cuerpo humano que requieren de la interrelación de los subsistemas para subsistir, como por ejemplo el aparato digestivo.

#### **El Usuario.**

Es la persona que de alguna manera utiliza un sistema para dar o recibir información, sin ser el sistema necesariamente automatizado. Esta persona generalmente conoce cuál es el procedimiento que debe seguir una actividad para generar el resultado deseado, es decir, él tiene la experiencia necesaria para realizar sus tareas y sabe cuales son sus necesidades de información.<sup>22</sup>

Todos en algún momento asumimos el papel de *usuario* de un sistema, ya sea un sistema de información, de un programa, de una máquina, de un coche, etc.

El propósito del presente documento es el de proporcionar herramientas y bases para el *usuario* ya que él juega un papel preponderante en el ciclo de vida de un sistema.

El *usuario* sabe como operar su negocio aún sin la automatización del mismo, es por ello que los ingenieros de sistemas deben extraer dichos conocimientos para su próxima automatización. Los sistemas de información tienen como objetivo dar tratamiento y *producir* más información para la toma de decisiones, es por ello que el usuario *debe* definir al cien por ciento las necesidades de información que él desea.

---

<sup>22</sup> Nota del Autor: He llegado a este concepto en base a la experiencia.

De la definición de necesidades (que se convierten en los requerimientos del sistema) se hace el análisis para el desarrollo de la automatización del sistema de información. Por ello es muy importante que el *usuario* cuente con las herramientas de análisis necesarias para describir su negocio o la actividad que realice.

El usuario debe indicar al ingeniero de sistemas en que contexto se mueve el negocio, que interrelaciones tiene y con quién son realizadas. También debe apoyar en la descripción de la organización (su negocio), así como las funciones y procesos que realiza para el logro de sus objetivos.

Es importante mencionar que una de las funciones del ingeniero de sistemas es precisamente lo mencionado en el párrafo anterior; pero hay que considerar que resulta más fácil trabajar el análisis y diseño del sistema con un *usuario* "que hable el mismo idioma". Mi experiencia laboral me indica que se facilita más el trabajo de análisis con aquel *usuario* que ya hizo un previo análisis de su negocio.

No se trata de que los *usuarios* se conviertan en ingenieros de sistemas y que éstos últimos se vuelvan expertos del negocio (tal vez se hagan expertos con el tiempo como una consecuencia).

#### **Una Clasificación de los Usuarios .**

Podemos clasificar a los usuarios de un sistema automatizado de la siguiente manera.

- **Capturistas.** Son aquellas personas que pertenecen al departamento al cual apoya un sistema y digitan el insumo a sus diferentes módulos. Por lo regular son los responsables de la actualización de los datos.
- **Operadores.** Son aquellas personas que se encargan de la operación cotidiana del equipo en el cuál reside el sistema, se encuentran estos en el centro de cómputo y no pertenecen a ingeniería de sistemas. Sus funciones generalmente son las de asignar recursos del equipo a las diferentes tareas que deba realizar el sistema, como por ejemplo montar y asignar cintas e impresoras.
- **Administradores de Bases de Datos.** Son aquellas personas que son los responsables de la integridad de la información, así mismo se encargan de monitorear los procesos del sistema. Generalmente controlan de acceso al sistema, los respaldos de la base de datos y la distribución de la información.

- **Analistas.** Son las personas del departamento que reciben los resultados del procesamiento de los datos, convertidos así en información, a la cual le dan un tratamiento específico, que a su vez ellos mismos pueden usar consecuentemente como herramienta de trabajo.
- **Estrategas.** Son las personas que reciben la información ya procesada para la toma de decisiones.

### *El Ingeniero de Sistemas.*

¿Por qué Ingeniero en Sistemas?, lo nombro así porque deseo ocupar el término de Ingeniería para definir al proceso de desarrollar un sistema y que es semejante al proceso de construcción de un edificio. Me refiero a CONSTRUCCION debido a que este término engloba a todo el proceso que origina un sistema.

El *Ingeniero de Sistemas* es la persona responsable de la construcción y el mantenimiento de un sistema.

En la actualidad en nuestro país un *Ingeniero de Sistemas* puede desempeñar diferentes roles en su carrera, incluso en paralelo; como un ejemplo mencionaré algunos de estos roles:

- Líder del Proyecto.
- Analista del Negocio.
- Diseñador.
- Administrador de la Base de Datos.
- Programador.
- Soporte Técnico.
- Capacitación.
- Capturista.

## PARTE II. LA PARTICIPACIÓN DEL USUARIO.

Para comenzar con este tema quisiera preguntarme *¿por qué se construye un sistema?*

Un sistema es construido debido a que se ha detectado una *necesidad* de una o varias personas a las cuales ya nombramos *usuarios*.

Algunos ejemplos son los siguientes :

- Crear un mecanismo que apoye en el procesamiento de información y que asegure la integridad de la misma.
- Reemplazar a un sistema ya existente por obsolescencia.
- Reemplazar a un sistema ya existente del cual no se obtienen los resultados deseados.

Los usuarios requieren que su información sea manipulada de manera eficiente y que se obtengan los resultados que ellos desean.

Los resultados de un sistema los podemos clasificar en:

1. Consultas interactivas en pantalla.
2. Reportes impresos.
3. Gráficas.
4. Microfichas.
5. Cintas.
6. Archivos.
7. Actualizaciones a la base de datos.

¿Qué puntos neurálgicos podríamos considerar que sirven como base para la construcción de un sistema?

## *EL USUARIO Y LA CONSTRUCCION DE UN SISTEMA Los Sistemas de Información*

### *El Conocimiento y la Experiencia.*

No me refiero al conocimiento y la experiencia de los usuarios e ingenieros en el área de sistemas; por otro lado, me refiero al conocimiento y la experiencia que del negocio tengan los Usuarios.

*El negocio de los Ingenieros es la construcción del sistema, el negocio de los Usuarios es el análisis, interpretación y toma de decisiones en base a los resultados que se obtengan del sistema.*

Los Usuarios conocen y usan cotidianamente las reglas del negocio y de integridad de la información que operan en un sistema para obtener productos los cuales explotan para la toma de decisiones.

*De su conocimiento y experiencia se origina el sistema.*

Probablemente los Ingenieros con el tiempo se hagan expertos en el negocio de los Usuarios y no es uno de los objetivos de la construcción de un sistema, así como también no es un objetivo el convertir a los Usuarios en Ingenieros de Sistemas .

Ahora bien,

*¿Quién comúnmente ya sea de forma manual y/o automatizada genera la información?*

*¿Quién comúnmente toma decisiones en base a la información generada?*

*¿Quién aprende y conoce del negocio en base a sus datos?*

*¿Quién es el receptor de la información del medio ambiente de la empresa?*

La respuesta a todas estas preguntas es: *EL USUARIO*

Un Ingeniero de Sistemas Profesional:

- No debe dar nada por hecho.
- No es un Sabelotodo.

- No debe devaluar o degradar las tareas o funciones que realice el usuario y/o la empresa.
- No debe inferir productos, procesos, funciones, etc.
- No es arrogante.
- No debe considerar ni tratar al Usuario como un tonto.

Del resultado de la interacción entre el Usuario y el Ingeniero del Sistemas dependen los productos del sistema. Al hablar de participación del Usuario en el proceso de construcción de un sistema me refiero al *esfuerzo y dedicación* que él le proporcione al mismo.

El Usuario no debe dejar al libre albedrío del Ingeniero de Sistemas todas las fases del proceso, ya que se pierde el control del avance del proyecto y se corre el riesgo de que se haya mal interpretado algún proceso de la operación del sistema.

Para iniciar con el proceso de construcción debe haberse *adquirido un compromiso* entre los Ingenieros de Sistemas y el personal del departamento Usuario. Lo anterior tiende a parecer un detalle muy insignificante y como tal muchas personas lo pasan por alto.

Este compromiso debe abarcar el respeto a los requerimientos, entrevistas, juntas, y resultados que se deriven del sistema. El no dedicarle el tiempo suficiente en cualesquiera que sea la fase de la construcción<sup>23</sup> puede traer como consecuencia un resultado no deseado.

Podemos concluir entonces que:

*Dar más énfasis y dedicación en la interacción entre los Usuarios e Ingenieros de Sistemas en todas las etapas del proceso de construcción de un sistema trae como consecuencia que los productos resultantes sean de la entera satisfacción del Usuario, derivándose como consecuencia que el mantenimiento del sistema se minimice.*

Otros factores que se ven afectados por este esfuerzo y dedicación en la interacción son:

---

<sup>23</sup> Las fases del proceso de construcción de un sistema serán explicados en el próximo capítulo.

- Menos Costo.
- Menos tiempo.
- Oportunidad de ocupar a los Ingenieros en otros sistemas.

### PARTE III. PLANEACION ESTRATEGICA DE SISTEMAS (BSP).

La *Planeación Estratégica de Sistemas* (Business Systems Planning) es una técnica estructurada para apoyar a las empresas a establecer un plan de Sistemas de Información que a largo plazo satisfaga las necesidades de información de la entidad.<sup>24</sup>

El creciente movimiento empresarial determinado por el medio ambiente, hace necesario que los ejecutivos de la misma tengan información al día veraz y oportuna. Otro fenómeno que ocurre en la actualidad es que *las empresas están reconociendo a los datos como un recurso más*, tan importante como los recursos humanos, materiales y financieros.<sup>25</sup>

Se puede reconocer que es necesario que se construyan Sistemas de Información que atiendan a la empresa en su totalidad, aunque en la definición del sistema pueden cometerse algunos errores como los siguientes:

- Fallas al obtener el compromiso y la participación del usuario.
- Establecimiento de objetivos y estrategias aisladas que no cumplan con los objetivos empresariales.
- Intento de implementar Sistemas de Información sin la previa comprensión del negocio desde un punto de vista de administración global.

La Planeación Estratégica de Sistemas fue creada para apoyar a la creación de un *plan empresarial* a través de :

1. Un análisis Top-Down (de arriba a abajo) para reunir al personal involucrado (iniciando con la administración del negocio y trabajando posteriormente con los niveles inferiores de la empresa) y mediante el estudio de la organización (trabajando en toda la entidad a nivel de detalle.
2. Un esfuerzo de implementación Bottom-Up (de abajo hacia arriba).
3. Uso de una metodología estructurada.

---

<sup>24</sup> Fuente: Business Systems Planning; Information Systems Planning Guide; IBM; 1990; U.S.A.

<sup>25</sup> Ídem.

4. La traducción de los *objetivos del negocio en requerimientos de información.*

BSP nace del programa establecido por IBM en 1970 después de haber experimentado en su propia empresa problemas de definición de datos.<sup>26</sup>

*Objetivos de la Planeación Estrategica de Sistemas.*

El primer y más importante *objetivo es generar un Plan de Sistemas de Información en la empresa que soporte los requerimientos de información a corto y largo plazo y su integración con el Plan Empresarial.*

Entre otros objetivos se encuentran los siguientes:

- Dar un método objetivo y formal para el establecimiento de las prioridades de los sistemas de Información.
- Dar protección financiera para los sistemas que se desarrollarán en el largo plazo.
- Crear relación laboral entre Ingeniería de Sistemas y los Usuarios.
- Identificar a los datos como un *recurso corporativo* que puede ser sujeto de planeación, administración y control de tal forma que pueda usarse con eficiencia por todo el personal de la empresa.

*Beneficios Potenciales.*

Los beneficios que trae consigo el uso de la metodología pueden dirigirse a tres grupos pertenecientes al área de administración:

*A la Alta Dirección.*

- Una evaluación de la efectividad de los sistemas de información actuales.
- Una idea concreta y lógica para ayudar a la solución de los problemas administrativos desde una perspectiva del negocio.
- Un panorama de las futuras necesidades de información basadas en los impactos que recibe el negocio y las prioridades establecidas.

---

<sup>26</sup> Ídem.

- Sistemas de Información que son relativamente independientes a la estructura de la organización.

*A la Administración Funcional y Operacional.*

- Una idea lógica y concreta para ayudar a resolver problemas de operación y de control administrativo.
- Datos consistentes a ser usados y compartidos por todos los usuarios.
- Participación en conjunto con la Alta Dirección para el establecimiento de los objetivos y directriz organizacional, así como de las prioridades de los sistemas.
- Creación de sistemas que son administrados y orientados al negocio mas que al procesamiento de datos.

*A la Administración de los Sistemas de Información.*

- Comunicación inter-empresarial.
- Personal mejor capacitado y mas experimentado en el procesamiento de datos para responder las necesidades del negocio.
- Alta participación del usuario en la definición de prioridades y en si en todo el proceso de vida del sistema.

El plan que resulte del estudio de sistemas no deberá considerarse como un ente estático y como lo mejor, simplemente representa la mejor manera de pensar en cierto tiempo y bajo las condiciones ambientales que dominen a la empresa en ese entonces.<sup>27</sup>

*El principal beneficio del estudio es que ofrece la oportunidad de crear un ambiente y un plan de acción inicial que permita al negocio reaccionar a los futuros cambios manteniendo y ajustándose a las prioridades y directriz empresarial, sin cambios radicales en el diseño del sistema; además de definir la función de sistemas de información para continuar con el proceso de planeación.*

---

<sup>27</sup> Ídem

#### **PARTE IV. CONCEPTOS BSP.<sup>28</sup>**

IBM concibe el inicio de la metodología como una necesidad de implantación de sistemas de información automatizados que se ajusten a los objetivos empresariales; pero podemos ir más allá aún, la metodología podemos concebirla sin necesidad de que el sistema de información este automatizado.

Las premisas que deben determinarse al inicio del estudio de BSP son:

- Un Sistema de Información ( S/I ) debe ir acorde a las metas y objetivos del negocio.
- Una estrategia de S/I debe cumplir con las necesidades de todos los niveles de la organización de la empresa.
- Un S/I dará consistencia en el uso de la información.
- Un S/I deberá ser flexible y capaz de sobrevivir a los cambios organizacionales y de administración.
- La estrategia de S/I deberá implementarse proyecto por proyecto para soportarlo en su totalidad.

La clave para el éxito en la planeación, desarrollo e implementación de una arquitectura de información es la siguiente:

1. Planeación Top-Down (De arriba a abajo) con una implementación Bottom-Up (De abajo hacia arriba).
2. Considerar y manipular a los datos como un recurso corporativo.
3. Orientación al negocio.

---

<sup>28</sup> El objetivo de esta parte de la investigación no es presentar la metodología en su totalidad, solo pretende mostrar los conceptos más importantes de ella. Para mayor información favor de referirse al manual de IBM BSP GE 20-0527-4.

## **Proceso del Estudio BSP.**

### **1. Compromiso.**

Un estudio BSP no debe iniciarse a menos que la alta dirección y algunos otros ejecutivos hayan hecho un compromiso común. El estudio debe reflejar su vista del negocio, y el éxito dependerá de su apoyo y de la definición de los requerimientos de información.

### **2. Preparación del Estudio.**

Todos los ejecutivos y el equipo de trabajo participantes *deberán conocer que harán, porque y que se espera de él.* Es conveniente que se capacite al personal y se le oriente, de tal forma que se reciba mejor retroalimentación del grupo de trabajo.

En esta fase del estudio deberá seleccionarse al personal que será entrevistado. La principal salida de esta fase será un documento que contenga el Plan de Trabajo, un Calendario de Entrevistas, y un Calendario de puntos a checar.

### **3. Inicio del Estudio.**

El estudio BSP requiere la completa participación de los miembros del equipo de trabajo (Ejecutivos, Usuarios, Ingenieros de Sistemas) e inicia con una reunión de revisión del negocio.

### **4. Definición de los Procesos del Negocio.**

Para el estudio BSP es muy importante el realizar las entrevistas con los ejecutivos, definiendo la arquitectura del negocio, los grupos de datos y las actividades que se realizan para ellos; cada individuo integrante del equipo debe comprender como funcionan los procesos que se llevan a cabo en la empresa.

El documento resultado de este paso es *la lista de los procesos, una descripción de cada uno y la identificación de aquellos que son claves en la vida del negocio.*

### **5. Definición de los Datos del Negocio.**

Para identificar los datos del negocio, es necesario haber identificado previamente las entidades y la información que usan.

En esta fase se deben agrupar los datos lógicamente por *categorías*. Esta clasificación ayuda en las siguientes fases del estudio e incluso en el desarrollo de las aplicaciones para eliminar redundancia.

#### **6. Definición de la Arquitectura de la Información.**

La definición de la arquitectura de la información implica *relacionar los procesos a las categorías de datos*.

Lo anterior permite la *evaluación del intercambio de información* dentro de la empresa.

#### **7. Análisis del Soporte a los Sistemas Actuales.**

El propósito principal de esta actividad es determinar cuantos procesos actuales soportan el negocio para considerarlos en lo futuro.

Se analizan los procesos del negocio, las aplicaciones y archivos actuales para identificar redundancias, ayudar a clarificar responsabilidades y para la completa comprensión del negocio.

La principal herramienta de análisis es la MATRIZ. La *matriz clave* para los entrevistadores es la de *procesos/organización*.

Esta herramienta no sólo preparará al equipo de trabajo para interactuar con los miembros de la empresa, también los ayudará a determinar las necesidades de información.

#### **8. Entrevistas a los Ejecutivos.**

Es una parte muy importante del análisis TOP-DOWN. Su propósito es validar el trabajo hecho por el equipo, determinar los objetivos, problemas, las necesidades de información y su valor; además del establecimiento de una relación e integración con los ejecutivos.

El resultado principal consiste en notas de las entrevistas, y diagramas del negocio.

### **9. Conclusiones.**

Algunos de los problemas del negocio fueron puestos como entrada durante el análisis del sistema, extendiéndolos y complementándolos por el conocimiento del equipo, y finalmente validados y explicados en las entrevistas a los ejecutivos.

Los problemas deben analizarse y relacionarse a los procesos del negocio para dar una guía y definir las prioridades del proyecto. También los problemas deben dividirse dentro de muchas categorías y habrá que concluir de ellos.

Los resultados de ésta tarea deben dar la base para desarrollar recomendaciones y para definir las prioridades entre los subsistemas de la arquitectura de información.

### **10. Definición de las Prioridades.**

El desarrollo y la implementación de una arquitectura de información no puede realizarse a la vez, el equipo de trabajo deberá establecer las prioridades de desarrollo de las aplicaciones y de las bases de datos.

Deberá decidirse que bases de datos deberán diseñarse e implementarse primero, además de establecer que aplicaciones se definirán en la fase siguiente.

Las prioridades se determinan creando una lista de las aplicaciones que darán forma al Sistema de Información, así se establece un criterio y se mide el impacto de las aplicaciones en base a él.

### **11. Desarrollo de Recomendaciones.**

El propósito de este paso es asistir administrativamente a los ejecutivos en sus decisiones sobre actividades del negocio por medio de recomendaciones para aquellos procesos que se consideren críticos.

El resultado de los pasos uno al once deberá conjuntarse en un documento para ser presentado a los ejecutivos y deberá incluir lo siguiente :

1. Un resumen ejecutivo.
2. El análisis en detalle conteniendo las descripciones de los procesos del negocio.

3. De ser posible deberá extraerse aquella información confidencial o aquel material que no debe ser publicado a las demás personas que participan en el estudio.

### **12. Compromiso.**

El compromiso inicia con la presentación del resultado del estudio preliminar a la alta dirección, donde se muestran *los objetivos, ventajas, resultados esperados y los recursos requeridos para realizar la función.*

## **CAPITULO III**

### **El Proceso de Construcción**

*... no le des el pescado en la boca, mejor enseñale a pescar!*

## PARTE I. AUDITORIA OPERACIONAL<sup>29</sup>.

El objetivo de esta fase es examinar los procesos del negocio en los cuales se plantean las necesidades de información. En este fase del análisis del negocio se deberá investigar, revisar y verificar los procesos de la empresa .

En la *auditoría operacional (AO)* se establecen el alcance, calendario y presupuesto estimado del proyecto, identificando los procesos que serán soportados por el sistema de información; estableciendo requerimientos, actos y fuentes de los hechos.

Los resultados de esta fase son *el presupuesto, la creación del macro-modelo de información y los modelos de actividades*. El grupo que aplica la auditoría operacional entrega sus conclusiones en un *informe final*, incluyendo comentarios sobre los aspectos de sus principales procesos y un *calendario* para la ejecución de las siguientes etapas de desarrollo.

Otro producto resultado de esta fase es la determinación de los *requerimientos de información* que tienen los usuarios.

*Es muy importante recordar la participación del usuario durante las reuniones de trabajo.*

### **1. Establecer el Alcance, Calendario y Presupuesto del Proyecto.**

En esta etapa se debe explicar qué es lo que va a ser auditado, cómo, cuándo y por qué. Los participantes deberán dar una explicación de las necesidades de información que están siendo auditados, además de explicar que áreas intervendrán en el proyecto.

Así mismo se deberá establecer el plan con el calendario para el proyecto. El plan identificará al personal involucrado en cada tarea y asignará la disponibilidad de recursos.

Todos los documentos resultantes de esta fase *deben ser revisados en conjunto con los usuarios y todo el personal involucrado*, para su aprobación o en su caso para su modificación.

---

<sup>29</sup> Metodología de Desarrollo de Sistemas Nuevo Enfoque; Banamex S.A.; Julio 1992



#### **4. Preparar las Entrevistas con el Usuario.**

Se requiere una preparación cuidadosa de los siguientes puntos para llevar a cabo las entrevistas con los usuarios.

1. Desarrollar una lista de usuarios a ser entrevistados, basada en la involucración de las unidades organizacionales con respecto a las actividades que se desarrollarán; se creará una lista de usuarios claves a ser entrevistados.

2. Construir un cuestionario: el grupo debe elaborar un cuestionario con preguntas como las siguientes.

2.1 Describa qué información llega a su departamento y el uso que se hace con ella.

2.2 Describa que información se produce en su departamento.

2.3 Describa los documentos que utiliza para producir la información descrita en el punto anterior.

2.4 Describa que información necesita y que actualmente no recibe.

2.5 Describa que información recibe actualmente y no le es útil.

El cuestionario debe buscar describir para cada actividad que información necesita y produce cada departamento de la empresa; así mismo debe captar las opiniones de los usuarios respecto a la utilidad y calidad de la información que necesitan y reciben.

#### **5. Documentación.<sup>30</sup>**

Se debe identificar quién es el responsable principal de los documentos y quién los tiene, así mismo con las políticas que afectan los procesos.

#### **6. Proceso de Análisis y Síntesis.<sup>31</sup>**

Una vez recopilada la información, el grupo encargado de la auditoría debe darse un tiempo para hacer el proceso de descomposición y de composición de la información.

---

<sup>30</sup> En la metodología original sus autores la llaman "Obtener copias de los Procedimientos, Documentos Fuentes, y Políticas que describen los procesos que serán auditados".

<sup>31</sup> No está determinado en la metodología original.

Es muy importante darnos cuenta que este proceso no es sólo de análisis ya que también al final deberá entregarse un informe con las posibles soluciones que se deriven del sistema de información.

### **7. Construcción del Macro Modelo de Información.**

Una vez terminado el proceso de análisis y síntesis de la información debe realizarse el macro modelo de información describiendo únicamente las entidades y las relaciones entre ellas (en este nivel aún no se especifican los atributos de las entidades).

El modelo debe ilustrar de una manera macro el primer intento de identificación de datos con la perspectiva de que se convierta en la base de datos del sistema de información.

Para realizar el modelo podemos hacer uso de la técnica de Entidad Relación Llave Atributo. El modelo debe ser mostrado y explicado a todos los participantes del proyecto para su discusión y posible aprobación.

### **8. Construcción de Modelos de Actividades para cada Proceso.**

En esta etapa se deben construir modelos de actividades para cada uno de los procesos del área a auditar. Estos modelos deben crearse siguiendo la técnica de análisis y diseño estructurado.

### **9. Producción del Reporte Final.**

El grupo de auditoría debe expresar sus conclusiones sobre los aspectos importantes que se hayan observado durante la auditoría. También se debe proporcionar soluciones a los problemas que se consideren críticos y que deban ser resueltos inmediatamente. Se deben definir claramente todos los problemas que se hayan detectado sean organizacionales, personales o de otra índole.

El grupo debe identificar las oportunidades que existen para mejorar la eficiencia del sistema y minimizar los costos.

Todos los diagramas, matrices y copias de formatos deberán ser incluidos, anexando los comentarios como apéndices de la auditoría.

## PARTE II. TECNICA DE ANALISIS Y DISEÑO ESTRUCTURADO. STRUCTURED ANALYSIS DESIGN TECHNIQUE "SADT".

SADT fue desarrollada a finales de 1960 durante la revolución de la programación estructurada<sup>32</sup>. Douglas T. Ross describió algunas ideas sobre SADT en 1969.

*Una actividad* es un procedimiento y/o función que suceden en la organización, todos con el objetivo de la generación de un producto.

Las actividades se relacionan una a otra por el uso común de información; ciertas actividades no pueden darse sin las entradas de información de otras actividades relacionadas y viceversa.

*Las actividades son eventos donde se procesa información.*

*Modelado.* Es el hecho de desarrollar una descripción certera de un sistema.

Un modelo SADT usa para la descripción de un sistema el lenguaje natural y la representación de ideas mediante símbolos.

Los modelos presentan las actividades en forma sucesiva detallada y define las relaciones entre las actividades.

Un modelo de actividades (también llamado *modelo de procesos*) es una serie de diagramas de cajas y flechas dibujadas jerárquicamente para representar eventos funcionales de la organización y el flujo de información entre los eventos.

El modelo es el primer paso para la producción de una estructura de información (arquitectura) para guiar a la creación de una base de datos y de las aplicaciones del sistema de información.

El propósito del modelo es responder a un conjunto de interrogantes; para SADT es muy importante definir los *límites del sistema*, que serán representados en el modelo, debido a que de ahí en adelante mostrarán que será cubierto por el y que no lo será.

---

<sup>32</sup> SADT, Structured Analysis Design Technique, pag 1.

El resultado final del proceso del estudio SADT es un conjunto de *descripciones lógicas* que inician desde la de mas alto nivel (de lo general) que muestra el *contexto* del sistema hasta las descripciones detalladas (lo particular) de la operación del sistema. Cada descripción es llamada *diagrama*, los modelos son organizados jerárquicamente en cuyo caso los diagramas del principio son menos detallados que los últimos. Los diagramas y los modelos tienen sus propias reglas de construcción, es muy importante conocer dichas reglas porque tienen un significado en la notación simbólica. Los diagramas SADT son capaces de definir sistemas con diferentes entornos y objetivos y puede mostrar como las actividades se interrelacionan.

Un diagrama contiene *cajas y flechas*. Las cajas representan actividades del sistema, las flechas conectan a las cajas para representar interfaces o interconexiones entre las actividades. Un diagrama es puesto en un formato, nombrado con un *título* puesto en la parte inferior central del formato; así mismo cada formato debe contener los siguientes datos de identificación: Autor, Proyecto, Fecha de Creación y de Última Modificación. Toda la información de identificación del diagrama es puesta en la parte superior de la forma.

Las cajas en un diagrama SADT son *rectángulos*. *Una caja representa una función o la parte activa de un sistema, así las cajas son nombradas con verbos.*

La técnica tiene una regla que sólo permite que *no haya menos de tres y no mas de seis cajas en cualquier diagrama*, es decir, las actividades no deben descomponerse en menos de tres ni tampoco en mas de seis actividades del siguiente nivel menor.

*En SADT cada lado del rectángulo indica algo en especial; del lado izquierdo de la caja se anotan las entradas, el lado derecho es reservado para las salidas, el lado superior es reservado para los controles y el lado inferior es reservado para los mecanismos.*

Proyecto	Autor	Fecha Revisión
Nodo	Título	Numero

Figura 2. Formato de Diagrama SADT.

Dicha notación representa ciertos principios de sistemas: *Las entradas son transformadas en salidas, los controles dictan bajo que condiciones se realizan las transformaciones y los mecanismos describen como se compone la función.*

En SADT las cajas nunca se colocan en el diagrama de manera aleatoria; son puestos de acuerdo al orden de importancia que juzga el autor del mismo. Así mismo las cajas deben ser *numeradas* para darles un identificador único y en base a ellas ordenar las actividades jerárquicamente.

Las *flechas* en un SADT son líneas con una punta al final de la misma; para un diagrama de actividades una flecha representa una colección de *objetos* como (planos, datos, máquinas e información). Adyacente a las flechas deberá ponerse frases que representen a los objetos, siendo estas frases concretas y representativas del objeto.

Existen cuatro posibles relaciones que los objetos pueden tener con las actividades: Servir como *Entradas*, *Salidas*, *Controles* y *Mecanismos*. Cada una de estas relaciones se representa con una flecha que conecta a un lado en particular de la caja.

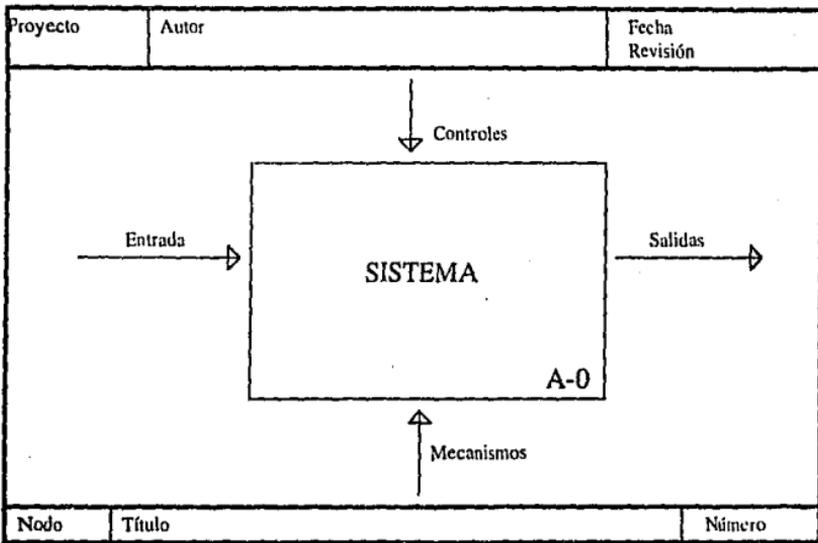


Figura 3. Representación de las Entradas, Salidas, Controles y Mecanismos.

Las *entradas* son aquellos objetos que serán transformados por las actividades, los *controles* son objetos que restringen a la actividad, las *salidas* son objetos que resultaron del proceso de transformación de las entradas, los *mecanismos* representan como se realizan las actividades, también representan los aspectos físicos de una actividad (por ejemplo lugares de almacenamiento, gente, organizaciones, dispositivos, etc.).

Por lo tanto tenemos que un diagrama SADT se compone de cajas interconectadas por flechas las cuales definen como una caja influye en otra; no son diagramas de flujo de procesos ni diagramas de flujo de datos.

En SADT se maneja el término de "jerárquico" debido a que un sistema puede ser representado en un diagrama por una sola caja, la división jerárquica de cada diagrama es un proceso llamado *descomposición*. La descomposición crea límites, cada caja detallada en un diagrama es considerado un límite formal alrededor del sistema que se describe.

En el proceso de descomposición cada diagrama subsecuente del *principal (padre)* es llamado "*diagrama hijo*".

El diagrama que consiste de una sola caja y de las flechas que definen sus entradas, salidas, controles y mecanismos, representando los límites del sistema es llamado "*Diagrama de Contexto*", el análisis en SADT inicia con el diagrama de contexto.

El proceso de modelado es una descomposición TOP-DOWN, es decir, de los *general a lo particular*. Dicho proceso inicia con el diagrama de contexto que a su vez será descompuesto en las subactividades que se requieren para realizar la actividad, luego entonces cada subactividad será descompuesta en las subactividades que contenga y así hasta llegar a las actividades en detalle. No hay que olvidar que existe la regla de tres y seis, la cual habrá de respetar.

Para numerar las actividades se sigue un proceso de composición numérica donde los se antepone el dígito del diagrama padre al descomponer el diagrama, así la actividad cero estará compuesta de las actividades uno, dos, tres, cuatro, cinco, seis (si es el caso). Luego los hijos de la actividad uno serán los uno uno, uno dos, uno tres etc., los hijos de la actividad cinco serán cinco uno, cinco dos, etc., los hijos de uno uno serán uno uno uno y así sucesivamente.

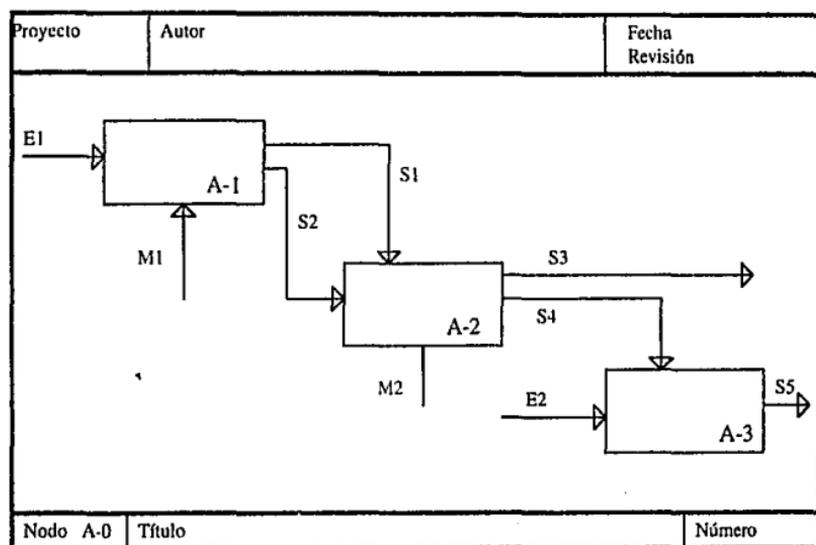


Figura 4. Representación del Análisis de Descomposición.

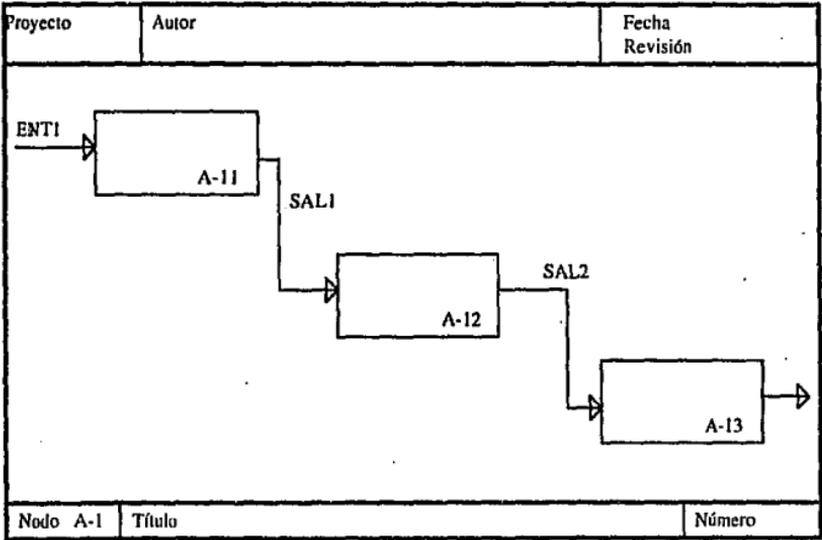


Figura 5. Representación de la numeración de actividades.

Cualquier metodología de análisis estructurado debe asegurarnos que todos los diagramas se conectan uno del otro con coherencia para hacer el modelo consistente; SADT maneja flechas externas, las cuales aparecen por encima fuera de los límites del diagrama, dichas flechas son la interfase entre el diagrama y el resto del modelo; SADT requiere que las flechas externas tengan relación con las internas para dar coherencia al modelo, en otras palabras se requiere que las flechas "embonen" unas con otras dentro y fuera de los límites del modelo.

La conexión de flechas asegura la descomposición consistente de las actividades.

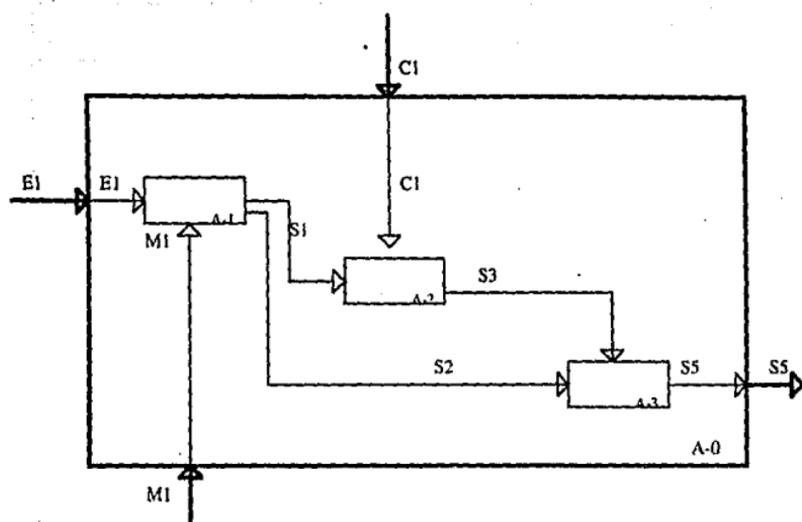


Figura 6. Representación de la conexión de flechas internas y externas.

La técnica de modelado tiene una base, ésta es el conocimiento del negocio donde se aplicará. SADT necesita de un proceso de entrevistas con los integrantes de la empresa para comenzar a trabajar.

Es necesario que el grupo que ejecuta el análisis se entreviste y que recabe todos los documentos que son necesarios para las actividades como facturas, pólizas, reportes etc., debido a que estas son las fuentes de información representadas como entradas, salidas, controles y mecanismos en SADT.

Es muy importante que se haga un compromiso con los ejecutivos de la empresa, debido a que se requiere una alta interacción de los usuarios con los Ingenieros de Sistemas, ya que se debe dar visto bueno de los modelos resultantes.

En el caso de que exista una auditoría administrativa o contable, nos podemos ayudar de ellas para realizar los modelos de actividades.

Al hacer una descripción de un sistema deben representarse sus objetos de tal forma que lleven un orden y que explique a su lector como trabaja en su medio ambiente.

#### **Túnel SADT.**

Hay ocasiones en todo tipo organizaciones<sup>33</sup> que se encuentran objetos y/o entidades con *destino y origen desconocido*, dichas entidades vienen y van de algún lado de las actividades, donde no es posible conectarlas con el resto de los diagramas, ni con el contexto del sistema. Este tipo de entidades son representados por flechas con puntas o con el inicio de la flecha encerrados entre paréntesis. Esta representación en SADT se le llama "Túnel".

Las puntas entre paréntesis indican que la información de dichos objetos va a alguna parte del modelo o bien que no serán tratadas en el resto del modelo. Los paréntesis que encierran al inicio de la flecha indican que la información proviene de alguna parte del modelo el cual no se puede determinar y no existe el interes de determinarlo.

#### **La diferencia entre Entrada y Control.**

Una de las grandes diferencias entre SADT y otras técnicas de diseño estructurado es el distinguir *las entradas de los controles*. Los controles permiten determinar las *restricciones* que deben ocurrir para el proceso de transformación de un insumo en producto. Las restricciones por lo tanto dan una imagen real de como debe operar el sistema para describir los hechos y las reglas que deben seguirse para la transformación.

#### **Los Mecanismos definen como las actividades son realizadas.**

En SADT un sistema es descrito desde una perspectiva funcional, dándole mucha importancia a los *mecanismos*. Los *mecanismos* permiten al analista definir con precisión como opera una función o proceso en particular y que se requiere usar para que se de y quien lo hará.

Con los *mecanismos* un analista puede agregar detalles al diagrama para no dejar nada en el "limbo" acerca de que sucede si se ejecuta la actividad.

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

---

<sup>33</sup> Ídem, pag 40.

Los *mecanismos* pueden usarse para diferentes propósitos; pueden usarse para identificar quien hará una función en particular. También pueden usarse para indicar que ciertas funciones soportan las operaciones de otras y por lo tanto deben suceder en una secuencia fija.

#### ***Retroalimentación de Control y Flujo de Datos.***

El concepto de retroalimentación es fundamental en la teoría de sistemas, sucede cuando la salida de una actividad afecta a la entrada de otra actividad; en SADT se pueden describir dos diferentes clases de retroalimentación: *La de Controles y la de Flujo de Datos.*

Distinguir entre la retroalimentación de *controles y de flujo de datos* es fundamental. La retroalimentación del *flujo de datos* sucede cuando la salida de una actividad se convierte en la entrada de otra. La retroalimentación de *Control* sucede cuando la salida de una actividad restringe u otra actividad para que suceda.

#### ***El Lenguaje Natural en SADT.***

SADT por si mismo nos apoya para dar una descripción del sistema por medio del *lenguaje natural* que nosotros manejamos y por la representación gráfica de las actividades. El *lenguaje natural* permite al Ingeniero de Sistemas expresar libremente la operación del sistema en términos familiares y entendibles para los usuarios.

Así el analista, durante el proceso de diagramación, toma las palabras del experto y les da una interpretación y significado en el lenguaje gráfico de SADT. Con el proceso de diagramación y la descomposición se logra eliminar la redundancia que exista en la interpretación del lenguaje natural del los expertos.

#### ***El proceso de Descomposición.***

La *descomposición* es el proceso de desarrollar un diagrama el cual detalle una caja (actividad) en particular y sus flechas asociadas. La *descomposición* implica usar *análisis* y *síntesis*, es decir, romper la actividad en pequeñas piezas y entonces combinarlas dentro de una descripción detallada de ese objeto. Es interesante notar que un modelo resultante muestra el resultado del juego entre análisis y síntesis.

Una buena técnica de apoyo al proceso de *descomposición* es elaborar una *lista de datos* y una *lista de actividades* e ir modelando de acuerdo a ambas listas.

Existen algunas estrategias de *descomposición*, dentro de ellas podemos encontrar la *descomposición funcional*, la *descomposición por rol del personal* o *por rol de la organización*.

***Descomposición Funcional.***

La *descomposición funcional* se considera la mejor, porque fuerza al analista a pensar acerca de *que hace* el sistema. De cualquier manera hay veces en que la *descomposición funcional* no es la mas adecuada para representar el modelo.

***Descomposición por Rol del Personal.***

La *descomposición por rol del personal* o *de la empresa* puede usarse como un principio para desarrollar descripciones de sistemas que documenten como la gente se coordina con otra mientras trabajan. Este tipo de *descomposición* se recomienda usarla solo al principio del modelado de toda la empresa.

***La precisión determina cuando se detiene la descomposición.***

La constante pero controlada evolución que llevan los diagramas en un modelo es un ejemplo de un proceso de ingeniería. SADT es un proceso de ingeniería porque permite el refinamiento sucesivo de una descripción del sistema; pero como cualquier otra actividad, el trabajo de refinamiento siempre debe detenerse en algún punto.

Los modelos SADT son construidos no sólo para responder a un conjunto específico de preguntas, sino también para responder a aquellas preguntas a un nivel dado de precisión. Así la *descomposición* se detiene cuando los diagramas en cualquier nivel son suficientemente detallados para reunir el propósito del modelo.

### PARTE III. TECNICA PARA EL DISEÑO CONCEPTUAL.

*El objetivo fundamental de esta fase es construir el esquema conceptual detallado de la base de datos a través del modelo de información.<sup>34</sup> Así mismo habrá que definir los procesos de actualización, mantenimiento y productos de la base de datos.*

*Esta es la última fase en la cual participa el usuario de manera activa en el desarrollo del sistema de información. En las próximas fases de la construcción del sistema, que son el diseño físico, construcción de aplicaciones, pruebas e implementación; la función del usuario será revisar y dar visto bueno (en su caso) a los procesos y productos del sistema de información, dichas fases pasaran a ser sólo responsabilidad del grupo de Ingenieros de Sistemas.*

*Cabe recordar que la presente tesis no pretende abordar el diseño físico, construcción de aplicaciones, pruebas e implementación de un sistema.*

#### *1. Desarrollo de las Vistas del Usuario.*

Las *Vistas de los usuarios* son las formas en que los usuarios desean que se les otorgue la información. Las vistas representan la base para la generación de todos los productos del sistema, por ejemplo, reportes, consultas, gráficas, archivos de interfase, etc.

Es muy importante tratar que el sistema de información nos de la posibilidad de obtener vistas sin necesidad de crear procesos extras para generarlas. A esta situación se le conoce en Sistemas como independencia de los datos de los procesos.

El origen de las vistas de los usuarios son los modelos de actividades que describen los procesos de obtención de información. Otras vistas del usuario provienen de los documentos obtenidos en la fase de Auditoría Operacional y de las entrevistas.

#### *2. Refinamiento del Modelo de Información.*

El grupo debe modificar y re-definir (en su caso) el modelo de información, basándose en los resultados de la investigación anterior.

---

<sup>34</sup> Metodología de Desarrollo de Sistemas Nuevo Enfoque; Sistemas Banamex; Julio 1992

**3. Identificar los Requerimientos de Entrada.**

En este paso el grupo debe identificar las entradas a la base de datos, para esto se deben realizar las siguientes tareas :

- Identificar las fuentes potenciales de información que entrará a la base de datos, sean datos, documentos o cualquier otro medio para ingresar información.
- Definir la frecuencia probable de actualización de los datos, especificando si son diarias, semanales, mensuales, anuales, etc.
- Identificar el volumen esperado de actualización a las bases de datos por tipo de transacción (altas, bajas y cambios) .
- Definir los requerimientos de seguridad para las actualizaciones.

Requerimientos de Entrada						
Entrada	Fuentes Potenciales	Frec. de Act.	Volumen de Act.	Formas de Entrada	Reqs de Seg.	Reqs SW Y HW

Figura 7. Formato de Requerimientos de Entrada.

- Definir cuáles de las actualizaciones afectarán a otras estructuras de información, controlando así la integridad referencial entre los datos de diferentes entidades.
- Identificar los requerimientos de hardware y software.

**4. Identificar los Requerimientos de Aplicación.**

Analizando los modelos de actividades, el grupo debe desarrollar los requerimientos de aplicaciones, es decir, todos aquellos reportes, pantallas o cualquier tipo de extracción de información de la base de datos.

Los formatos en detalle de cada reporte aún no deben ser especificado, eso se realiza en fases posteriores.

El grupo debe :

- Identificar las Unidades Organizacionales a donde se irá la información, es decir, los organismos a los que pertenecen las personas que usarán la información.

Requerimientos de Aplicacion							
Aplicacion	Deptos u Oficinas	Frec. Est. de Salidas	Vol. Est. de Salidas	Reqs. de Seg.	Formas de Salida	Propiedad de la Aplic.	

Figura 8. Formato de Requerimientos de Aplicación.

- Identificar el volumen y frecuencia estimada de las salidas.
- Identificar los requerimientos de seguridad para las aplicaciones que sean necesarias.
- Definir las diferentes formas de salida como gráficas, archivos, discos, reportes y listados, consultas, etc.
- Identificar prioridades respecto a las aplicaciones a desarrollar, se deberá establecer dos categorías de aplicaciones, las "indispensables" y las "deseables", con base en la utilidad prevista de las aplicaciones.

5. Construir la Matriz de Desarrollo.

En este punto habrá de definir las relaciones entre el modelo de información y los requerimientos de aplicación utilizando un formato establecido.

Los requerimientos de aplicación que son prioritarios establecerán una pauta en la estrategia de desarrollo.

REQUERIMIENTOS DE APLICACIONES :												
MODELO DE INFORMACION	URGENTES					MAS				DESEABLES		
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
ENTIDAD A		X	X				X					X
ENTIDAD B	X		X		X		X		X	X		X
ENTIDAD C		X		X		X		X				

Figura 9. Matriz de Desarrollo.

## **6. Producir el Reporte Final.**

### **- Resumen de las propuestas.**

Se presenta un resumen de las propuestas de diseño, deberá incluir una descripción del esquema conceptual de la base de datos y de las estrategias recomendadas para el desarrollo.

### **- Describir los beneficios que se obtendrán.**

Se debe explicar cómo se piensa solucionar los problemas expuestos en el reporte de la Auditoría Operacional. Se identificarán las áreas en donde se incrementa la eficiencia y las de bajo costo. Se deben enfatizar los beneficios que se obtendrán al resolver los problemas existentes en el sistema.

### **- Creación del Plan de Desarrollo.**

El grupo deberá hacer un estimado del tiempo y recursos que serán necesarios para el desarrollo del sistema de información; dicha labor la deberá hacer solamente el grupo de Ingeniería de Sistemas.

Se deben describir los problemas potenciales generados por el plan de desarrollo.

### **Productos a Obtener al Final de la Fase del Diseño Lógico .**

1. Alcance, Calendario y Presupuesto del Diseño Conceptual.
2. Modelo de Información descrito a detalle.
3. Reglas de Integridad.
4. Reglas del Negocio.
5. Requerimientos de Entrada
6. Requerimientos de Aplicación.
7. Matriz de Desarrollo.

**8. Plan de Desarrollo.**

**9. Reporte Final .**

Como mencione anteriormente, el diseño conceptual es la última fase en la que participa el usuario antes de la construcción del sistema. Como usuarios debemos dejar que los expertos en sistemas construyan la aplicación. Nuestra siguiente participación activa será en la fase de la *Prueba Operativa*.

#### PARTE IV. PRUEBA OPERATIVA.

El objetivo de esta fase es *probar* la base de datos y las aplicaciones en un ambiente similar al real para *evaluar* ciertos factores como tiempo de respuesta y utilización de los recursos, además de finalizar la documentación del sistema.

La base de datos y las aplicaciones construidas se ejecutan en paralelo (si ya existen otras), probando y evaluando la ejecución del sistema para que los usuarios continúen familiarizándose en las mismas antes de su liberación.

Aquí debe tenerse cuidado de no liberar la aplicación antes de aplicarle la *Revisión de la Calidad del Sistema*.

Sus fases son las siguientes :

##### 1. Preparar el Plan de la Prueba Operativa.

Se deben detallar las tareas y recursos involucrados en la Prueba, recordando que no se permiten cambios a la base de datos ni a la aplicación, *debido a que se va a evaluar*.

##### 2. Llevar a cabo pruebas de Integración de Sistemas.

En esta fase se preparan las pruebas de la base de datos, sus rutinas de mantenimiento, sus aplicaciones y sus funciones críticas. Además se hace necesario que se hagan las pruebas necesarias acerca de la integración de la aplicación con otros sistemas.

Las pruebas deben considerarse formales, el grupo debe usar monitores de ejecución, generadores de datos de prueba y hacer un "*benchmark*"<sup>35</sup>. Así mismo deberán considerarse *Pruebas de Funcionalidad, de Eficiencia y de Mantenimiento*. En estas pruebas deben establecerse casos, archivos y transacciones de prueba, conociendo la salida correcta y los rangos de aceptabilidad de los resultados.

##### 3. Terminar la Documentación del Sistema.

Se deberá terminar la *documentación de operación y técnica*, necesaria para llevar a cabo las pruebas de aceptación y en paralelo. Es muy importante también establecer los controles para las modificaciones al sistema.

---

<sup>35</sup> Un Benchmark es una evaluación que sirve para comparar el comportamiento de objeto en base a otros.

El *Manual de Operación* debe contener al menos lo siguiente :

- Flujo de Operación.
- Descripción de la Base de Datos.
- Descripción de Mensajes.
- Descripción de Reportes.
- Posibles soluciones en caso de error.

El *Manual Técnico* debe contener al menos lo siguiente :

- Modelo de Actividades.
- Matriz de Procesos vs. Unidades Organizacionales.
- Vistas de Usuario.
- Modelo de Información detallado.
- Matriz de Desarrollo.
- Diccionario de Datos.
- Estructura de la Base de Datos.

#### **4. Probar la Base de Datos en un Ambiente similar al Real.**

En esta fase la base de datos es preparada para funcionar en un medio ambiente lo mas cercano al real, con datos reales y actuales del sistema actual automatizado o no.

Habrà que inicializar la Base de Datos con datos reales y recientes obtenidos de la fuente original, o bien de cintas, discos, etc. También hay que evaluar la integridad de las entradas. Luego entonces hay que evaluar el uso de la Base de Datos, analizando las entradas y las salidas en su ambiente de operación. Posteriormente hay que evaluar y documentar los calendarios de producción y las responsabilidades del personal.

**5. Entregar la Documentación al Usuario.**

Hay que hacer la entrega formal de la documentación al usuario, previendo que se hagan las modificaciones previas pertinentes al documento.

**Productos al Final de la Fase.**

1. Plan de la Prueba Operativa.
2. Plan de Pruebas Integrales y en Paralelo.
3. Reporte de Pruebas.
4. Manual de Operación y Manual Técnico.
5. Documentación al Usuario.

Considero importante hacer notar que aquí aún no se libera el sistema, la liberación formal se da después de que otro grupo (si es posible, si no el mismo grupo que construyo la aplicación) evalúa y certifica la calidad del sistema.

*La certificación del sistema debe hacerla un grupo experimentado de Ingenieros de Sistemas, el cual dará su fallo para la posible liberación de la aplicación a producción.*

## **CAPITULO IV**

### **Técnicas de Apoyo**

*...lo importante no es competir, lo importante es ganar!  
...ganar no lo es todo, es lo único!*

## PARTE I. ADMINISTRACION DE DATOS.

La *administración de datos* es la disciplina que se encarga de mejorar la calidad de la información para la toma de decisiones dentro de una empresa.<sup>36</sup>

Lo anterior implica que con esta disciplina se debe facilitar la planeación, obtención, clasificación, almacenamiento, control y seguridad de los datos a partir de los cuales se genera la información .

Las principales *Estrategias* de la *administración de datos* son:

- Identificar y definir los datos corporativos que se requieren en la empresa.
- Definir y manejar los modelos de datos que faciliten el desarrollo de nuevos sistemas.
- Organizar los datos de forma tal que puedan ser compartidos e independientes de las aplicaciones que requieran de ellos.

Los *BENEFICIOS* que se pueden obtener con esta disciplina serían:

- Incremento en la calidad de la información, debido al incremento en la precisión y confiabilidad de los datos.
- Permite la estandarización en la nomenclatura y características de los datos.
- Manejo consistente de los datos al identificar la reglas de integridad, las reglas del negocio y las relaciones entre las entidades y los datos.
- Se obtiene la independencia de datos pues no estarán ligados a las aplicaciones en forma de acceso ni en organización física.
- Reducción de costos de mantenimiento, al tener control de los datos.

Las *Funciones* que hay dentro de la administración de datos son:

- Planeación

---

<sup>36</sup> Fuente: Apuntes personales; Conferencia de Nomenclatura de Datos; Banamex S.A.; 1990.

- Diseño Lógico
- Certificación de Calidad de los datos
- Capacitación

#### **Planeación.**

En ésta fase debemos realizar el proceso de definición de los datos y su análisis dentro del ambiente de la empresa.

Para ejecutar la tarea anterior debemos contemplar el contexto<sup>37</sup> empresarial, tomando en cuenta los insumos, procesos y productos.

#### **Diseño Lógico.**

El propósito de esta fase es *desarrollar modelos de información* por área del negocio. Dichos modelos apoyarán el análisis y la comprensión de los datos. Dentro de estos modelos de información se identifican las entidades y sus atributos, las relaciones entre ellas, las llaves por entidad, los procesos, funciones, actividades, así como las reglas del negocio y las de integridad; es aquí donde se aplican los estándares de nomenclatura de datos que se hayan definido en la empresa.

#### **Certificación de la Calidad de los Datos.**

Una vez que hemos definido nuestros datos de acuerdo a nuestra nomenclatura, es necesario hacer una labor de certificación de calidad de los datos. Lo anterior implica analizar en conjunto la definición, integridad, validez y consistencia de los datos contando con la ayuda del departamento usuario. Esta labor es muy importante para la empresa ya que de su resultado depende la futura manipulación de los datos.

#### **Capacitación.**

Al haber certificado los datos la última tarea que nos queda por hacer es capacitar tanto al personal de sistemas responsable del desarrollo así como al personal usuario. Se les debe hacer de su conocimiento los estándares de la nomenclatura de datos y cual fue el resultado, todo esto con el motivo de que se haga oficial su uso en la empresa.

---

<sup>37</sup> Al referirme al Contexto, hablo de las interrelaciones de una empresa con su medio ambiente interno y externo.

3. Responsabilizar a los usuarios de la asignación de Claves de Usuarios. No dejar al departamento de sistemas como responsable, ni con acceso a la modificación de la seguridad.
4. Identificar grupos de usuarios por función y departamento usuario.
5. Identificar usuarios por tarea.
6. Crear Claves por Funciones, Grupos y por Usuario.

Por ejemplo. Existe un sistema que opera la Administración y Contabilidad de una empresa, el cual deben usar los departamentos de Personal y de Contabilidad propiamente.

El sistema tiene un módulo de captura de movimientos contables y un módulo de actualización de la cuenta de impuestos por pagar. Así asignamos claves por usuario de tal manera que la gente de personal no pueda modificar otra cuenta que no sea impuestos por pagar y de manera recíproca la gente de contabilidad no pueda modificar la nómina.

El concepto de seguridad no solamente se aplica al acceso de información dentro del ambiente de computación, también se aplica para el uso de los equipos periféricos conectados a un host y para la ejecución de los procesos de un sistema.

**PARTE IV. ELKA<sup>44</sup>.**

La técnica de modelado de Información *Entity Link Key Attribute (ELKA)* Entidad Relación Llave Atributo fue desarrollada por la Compañía Hughes Aircraft como parte de una metodología de uso general para la construcción de modelos de ingeniería y sistemas de manufactura.

*El objetivo principal de ELKA es crear un modelo de información del negocio con tendencia a la creación de una base de datos de la institución.*

**Premisas para el desarrollo de ELKA.**

1. La técnica debe definir completamente de una manera gráfica y de fácil comprensión el uso de los datos de la organización.
2. El modelo debe ser descrito usando una técnica formal descriptiva, en términos matemáticos de teoría de conjuntos.
3. El modelo resultante deberá contener toda la información necesaria para representar el sistema actual, basado en el análisis previo de documentos y en entrevistas.
4. El modelo resultante deberá ser claro y fácil de entender aún para usuarios no orientados a la computación.

Los conceptos de entidad, llave, atributo fueron explicados en el capítulo I, el concepto de relación también lo fue, pero aquí hay que definir los *tipos de relación que existen*.

Las entidades son representadas por cuadros, los atributos se anotan en forma de columnas dentro de los cuadros; las llaves son encerradas entre paréntesis o bien subrayadas; las relaciones se representan con líneas que tiene tres diferentes símbolos como terminadores:

---

<sup>44</sup> Elka Information Modeling; T.L. Ramey, R.R. Brown; Hughes Aircraft Company; El Segundo, California, U.S.A.

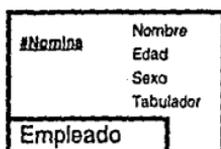


Figura 2. Representación de una entidad.

Cuando se inicia un modelo de información con ELKA debe hacerse usando la técnica de TOP-DOWN (de lo general a lo particular).

1. Generar una gran diagrama de contexto en el cual identifiquemos a todas las entidades, dando nombre a estas.
2. Complementar el diagrama anterior anotando las relaciones que se encuentren, utilizando la cardinalidad.
3. Identificar la llave primaria de cada entidad.
4. Establecer las llaves foráneas entre las entidades.

Una llave foránea es aquel atributo que se estableció como llave primaria para una entidad y que además es un atributo de otra entidad. Las llaves foráneas son de vital importancia, por que son las que nos apoyaran a identificar las relaciones entre las entidades.

5. Listar los atributos de cada entidad del modelo de información.

Es muy importante refinar el diagrama de contexto después de cada paso listado arriba.

#### Cardinalidad 1:1.

Interpretada de la siguiente manera, "de uno a uno", esto quiere decir que para una ocurrencia de la entidad "A" existirá una y solo una ocurrencia de la entidad "B".

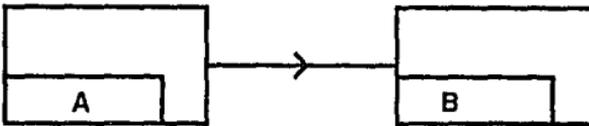


Figura 3. Representación de la cardinalidad 1:1.

Podemos mencionar algunos ejemplos:

En un sistema en el que se evalúa el servicio de prestación médica que una empresa otorga a sus empleados, encontramos la entidad empleado y derechohabiente y podemos establecer lo siguiente:

Un derechohabiente recibe el servicio porque lo afilia al mismo un empleado.

Así, determinamos como entidades al empleado y al derechohabiente, estableciendo la relación "afilia a", la cual se anota por encima de la línea, usando un signo "de mayor que".



Figura 4. Ejemplo de la cardinalidad 1:1

La notación podrá leerse así: Un empleado afilia a un derechohabiente

#### Cardinalidad 1:0:M

Interpretada de la siguiente manera, "de uno a cero, uno o muchos", es decir, que para una ocurrencia de la entidad "A" puede haber cero, una o muchas ocurrencias de la entidad "B".

Dicha relación se representa por medio de un "rombo en blanco".

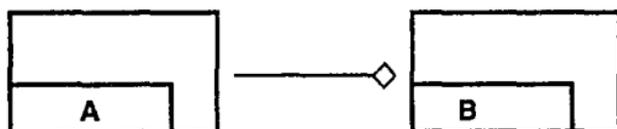


Figura 5. Representación de la cardinalidad 1:0:M.

Continuando con el ejemplo anterior, en el proceso de análisis nos podemos encontrar con la siguiente situación :

Un empleado puede tener cero, uno o mas familiares; aquí lo mas importante que debemos entender es que puede darse el hecho de que no los tenga, o que al menos tenga uno.

Dicha situación se representa de la siguiente manera.

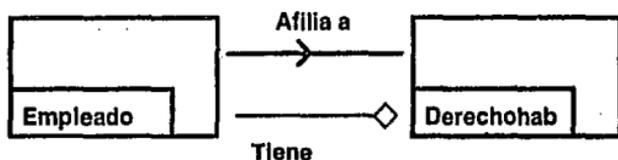


Figura 6. Representación las cardinalidades 1:1 y 1:0:M.

#### Cardinalidad 1:M.

Interpretada de la siguiente manera, "de uno a muchos forzoso", es decir para una ocurrencia de la entidad "A" existe una o muchas de la entidad "B". Esta relación se le conoce como forzosa, ya que para establecerla es necesario que se de al menos una ocurrencia de la entidad "A".

La cardinalidad 1:M es representada por un "rombo relleno".

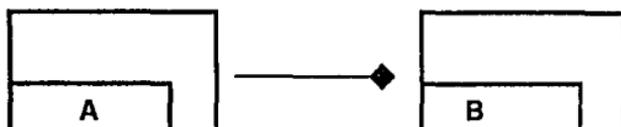


Figura 7. Representación de la cardinalidad 1:M.

Siguiendo con el mismo ejemplo de la prestación del servicio médico podemos encontrar que un médico puede dar consulta en más de un domicilio, pero forzosamente y al menos en uno. Dicho ejemplo puede ser representado de la siguiente manera.

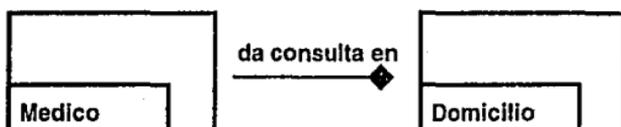


Figura 8. Ejemplo de la cardinalidad 1:M.

La cardinalidad M:N, es decir que para muchas ocurrencias de la entidad "A" existen muchas ocurrencias de la entidad "B", no es tomada en cuenta por el modelo ELKA, debido a que dicho modelo debe ser *normalizado*.

La normalización requiere de mucha experiencia y de un amplio conocimiento de la técnica, dejemos esa tarea al personal de Ingeniería de Sistemas.

Cuando el personal de análisis de datos es "novato", este tiende a encontrar que todas las relaciones entre entidades son M:N.

## PARTE V. DICCIONARIO DE DATOS.

Es necesario que dentro de nuestra organización existan el control necesario para el manejo de la información con la cual laboramos. El diccionario de datos es utilizado con y sin la automatización de la misma.

Este control es necesario ya que los Ingenieros de Sistemas, Programadores, Ejecutivos y demás Usuarios tendrán una idea clara de los datos que están disponibles en el sistema de información de la organización.

El diccionario de datos es una herramienta usada en la *administración de datos*, producto del establecimiento de los estándares de *nomenclatura de datos*.

Bajo el control del diccionario de datos, es factible tener información acerca del uso que se le da a los datos en el sistema de información. Otra de las utilidades que tiene el diccionario de datos es el servir como base para un mejor y más rápido diseño y mantenimiento de sistemas de información futuros.

Diccionario de Datos

Clase	Mod.	Entidad	Tipo	Long	Nmonico	Reglas de Integridad	Procesos en donde se usa
Nombre del Empleado			Cadena	30	Nom_Emp	< 5000 y > 500	Nomina y Cont
Numero de Cuenta			Entero	10	No_Cta		
Fecha de Ingreso del Empleado			Fecha	8	Fe_ing_Emp	Mayor al 1/Ene/90	Nomina

Figura 9. Ejemplo de Diccionario de Datos.

Long significa la longitud o tamaño del campo; Nmonico es el nombre con el cual se conocerá al campo dentro de toda la organización para efectos de su procesamiento.

## CONCLUSIONES

El presente trabajo ha mostrado una serie de técnicas que podrán servir de guías para los usuarios de los sistemas de información; como lo habra notado el lector no se ha insistido en usar una técnica como universal, sino más bien se ha tratado de integrarlas para definir un esquema global que pretende cubrir las diferentes etapas durante el proceso de construcción de un sistema.

Las herramientas aquí presentadas son las que en base a mi experiencia y conocimiento considero que pueden ayudar a los usuarios, con esto no quisiera atreverme a decir que son las mejores y que son universales, por el contrario, mi propósito fue tomar partes de diferentes técnicas y metodologías para proponer una manera de construir sistemas con la interacción del usuario.

La integración del usuario durante el proceso de construcción de un sistema es básica, debido a que él conoce, opera y administra su negocio, cualquiera que sea este. Dicha imagen del usuario debe ser transmitida en detalle al ingeniero en sistemas para que su labor sólo se remita a traducir la imagen en un sistema de información.

## Bibliografía

- Software Engineering : A practitioner's approach; Pressman, Roger S.; McGraw-Hill Series in S. Eng. & Tech; 1982; U.S.A.
- An Introduction to Database Systems; Vol I; Fifth Ed.; The Systems Programming Series; C.J. Date; Addison-Wesley Publishing Co.;1990, U.S.A.
- Data Base Management Systems, Second Edition; Alfonso F. Cardenas; Wm. C. Brown Publishers; 1989; Iowa, U.S.A.
- Estructura de Datos en Pascal; Aaron M. Tenenbaum, Moshe J. Augenstein; Printice Hall Hispanoamericana; 1985; México .
- Estructura de Datos y Diseño de Programas; Robert L. Kruse; Prentice Hall Hispanoamericana; 1985; México.
- Prototype Development Methodology for Databases; Seminar Outline; Computomata International Corp.; L.A California. U.S.A.; 1986
- Business Systems Planning.; Information Systems Planning Guide; I.B.M.; 1989; New York, U.S.A
- SADT, Structured Analysis and Design Technique; David A. Marca, Clement L. McGowan; McGraw Hill Book Company; 1986; U.S.A.
- Metodología de Desarrollo de Sistemas Nuevo Enfoque; Sistemas Banamex; Julio 1992; México D.F.
- Las Computadoras y la Información; Lawrence S. Orilia; Traductor Roberto Escalona García; 3a. Edición; Ed. Mc Graw Hill; 1988; México D.F.