

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

La administración de las bases de datos en las obras

Autor: Julieta Isabel Arriaga García

**Tesina presentada para obtener el título de:
Lic. En Sistemas computarizados [sic]**

**Nombre del asesor:
Sergio Francisco Barraza Ibarra**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





UQ
UNIVERSIDAD
VASCO DE QUIROGA

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

"LA ADMINISTRACIÓN DE LAS BASES DE DATOS EN LAS OBRAS
PÚBLICAS"

TESINA

Que para obtener el título de:

LICENCIADO EN SISTEMAS COMPUTARIZADOS

Presenta:

JULIETA ISABEL ARRIAGA GARCIA

Asesor:

ING. Y M.A. SERGIO FRANCISCO BARRAZA IBARRA
LIC. MANOELLA ATZIRY HERNÁNDEZ MONTIEL



CLAVE 16PSU0014Q
ACUERDO 952006

MORELIA, MICH.
MAYO 1999



UQ
UNIVERSIDAD
VASCO DE QUIROGA

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

"LA ADMINISTRACIÓN DE LAS BASES DE DATOS EN LAS OBRAS
PÚBLICAS"



Que para obtener el título de:

LICENCIADO EN SISTEMAS COMPUTARIZADOS

Presenta:

JULIETA ISABEL ARRIAGA GARCIA

Asesor:

ING. Y M.A. SERGIO FRANCISCO BARRAZA IBARRA
LIC. MANOELLA ATZIRY HERNÁNDEZ MONTIEL

CLAVE 16PSU0014Q
ACUERDO 952006

MORELIA, MICH.
MAYO 1999



UQ
UNIVERSIDAD
VASCO DE QUIROGA

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

"LA ADMINISTRACIÓN DE LAS BASES DE DATOS EN LAS OBRAS
PÚBLICAS"



Que para obtener el título de:

LICENCIADO EN SISTEMAS COMPUTARIZADOS

Presenta:

JULIETA ISABEL ARRIAGA GARCIA

Asesor:

ING. Y M.A. SERGIO FRANCISCO BARRAZA IBARRA
LIC. MANOELLA ATZIRY HERNÁNDEZ MONTIEL

CLAVE 16PSU0014Q
ACUERDO 952006

MORELIA, MICH.
MAYO 1999

A DIOS:

*Por darme la oportunidad de vivir,
y de rodearme de tanta gente que me quiere.*

GRACIAS SEÑOR.

A MIS PADRES † :

*Por guiarme desde donde estén,
y no dejarme sola en ningún momento.*

A MI PADRE † :

(DON PI)

*Si amo tanto la memoria de mi padre,
si no puedo separarme de su imagen,
si el tiempo no puede cortar el hilo
es porque con la experiencia,
comprendo todo lo que ha hecho por mí.*

A MI MADRE:

*A ti que con cariño y admiración,
por los innumerables esfuerzos que has hecho
para formarme como gente de provecho,
has tenido todo el coraje para enfrentarte a las
mil adversidades que se te han presentado en la
vida. poderte ofrecer este momento es una de mis
grandes satisfacciones en la vida.*

A MIS ABUELOS:

*Por apoyarme siempre que los
necesito y por darme ánimos para
seguir adelante.*

MIL GRACIAS.

A MIS TIOS:

*No tengo palabras para darles las gracias por
todo lo que han hecho por mí. y por quererme
como a una hija y como a una hermana. y
nunca dejarme a medias en este camino, y por
todo su apoyo.*

A MIS PRIMOS:

*Por ayudarme dándome apoyo
moral para no desistir, ya que
siempre estaremos unidos
por el amor.*

AL ING. SERGIO BARRAZA:

*Por su noble interés en alentarme
y orientarme a realizar mi gran anhelo.*

A LA LIC. ATZIRY HERNANDEZ:

*Por la forma tan entusiasta
en que me ha impulsado
a realizar este trabajo.*

A TODOS MIS COMPAÑEROS:

*Que dieron bellos recuerdos
a toda mi larga vida estudiantil.*

A MIS PRIMOS:

*Por ayudarme dándome apoyo
moral para no desistir, ya que
siempre estaremos unidos
por el amor.*

AL ING. SERGIO BARRAZA:

*Por su noble interés en alentarme
y orientarme a realizar mi gran anhelo.*

A LA LIC. ATZIRY HERNANDEZ:

*Por la forma tan entusiasta
en que me ha impulsado
a realizar este trabajo.*

A TODOS MIS COMPAÑEROS:

*Que dieron bellos recuerdos
a toda mi larga vida estudiantil.*

A MIS AMIGOS:

Por siempre estar a mi lado
en las buenas y en las malas,
por darme grandes palabras
de aliento y no dejarme sola.

MIL GRACIAS.

A MIS PADRINOS:

Para quienes sólo tengo palabras de
agradecimiento y respeto por su apoyo moral y
económico que me han brindado.

MIL GRACIAS.

A MIS MAESTROS:

A quienes hago presente
mi más cordial agradecimiento.

A TODOS:

Los que de una manera u otra han participado
en mi formación, mi más sincero reconocimiento
y profundo agradecimiento.

INDICE

1. INTRODUCCIÓN	1
2. ANTECEDENTES	4
3. OBJETIVO	7
4. ADMINISTRACIÓN DE BASES DE DATOS	8
4.1. DEFINICIÓN DE BASES DE DATOS	8
4.2. DISEÑO DE INTERACCIONES CON LA BASE DE DATOS	9
4.3. DESARROLLO DE SISTEMAS EN UN AMBIENTE DE BASE DE DATOS	10
4.3.1. RELACIONES ENTRE LOS DATOS	10
4.3.2. DESCRIPCIÓN DE LAS RELACIONES ENTRE ENTIDADES	12
4.3.2.1. DIAGRAMAS DE ESTRUCTURAS DE DATOS	14
4.3.3. COMPARTIR DATOS ENTRE LAS APLICACIONES	15
4.3.4. EL IMPACTO DE LOS SISTEMAS DE MANEJO DE UNA BASE DE DATOS EN EL DISEÑO DE SISTEMAS	16
4.3.5. MODELOS DE DATOS	20
4.3.6. ESTRUCTURACIÓN DE DATOS	22
4.3.7. NORMALIZACIÓN	22

5. MANIPULACIÓN DE DATOS CON UN SISTEMA DE ADMINISTRACIÓN -----	35
5.1. COMO SE USAN LOS DBMS -----	36
5.1.1. LOS DBMS -----	38
5.1.2. PROGRAMAS Y LOS DBMS -----	39
5.1.3. BASES DE DATOS INTERACTIVAS -----	40
5.2. FUNDAMENTOS DE BASES DE DATOS -----	40
5.2.1. ESTRUCTURA DE LAS BASES DE DATOS -----	41
5.2.2. ORDENAMIENTO DE LOS DATOS -----	48
5.2.3. FORMAS -----	51
5.2.4. ENLACE DE TABLAS DE BASES DE DATOS -----	55
5.3. INTEGRACIÓN CON LA BASE DE DATOS -----	59
5.3.1. NAVEGACIÓN, VISTA Y EDICIÓN DE REGISTROS -----	60
5.3.2. BUSQUEDA EN BASES DE DATOS -----	63
5.3.3. CREACIÓN DE INFORMES -----	67
5.4. PROGRAMAS QUE USAN BASES DE DATOS -----	67
5.4.1. USO DE BASES DE DATOS DESDE UN PROGRAMA -----	68
5.4.2. ESTÁNDAR xBASE -----	68
5.4.3. BASES DE DATOS DISTRIBUIDAS Y LA RELACIÓN CLIENTE - SERVIDOR -----	70
5.5. QUE ESPERAR EN EL FUTURO -----	72

6. CICLO DE VIDA -----	74
6.1. INVESTIGACIÓN PRELIMINAR -----	76
6.1.1. ACLARACIÓN DE LA SOLICITUD -----	77
6.1.2. ESTUDIO DE FACTIBILIDAD -----	77
6.1.3. APROBACIÓN DE LA SOLICITUD -----	78
6.2. DETERMINACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA -	79
6.3. DISEÑO DEL SISTEMA -----	80
6.4. DESARROLLO DE SOFTWARE -----	81
6.5. PRUEBA DE SISTEMA -----	82
6.6. IMPLANTACIÓN Y EVALUACIÓN -----	83
7. CONCLUSIONES -----	86
8. BIBLIOGRAFIA -----	87
9. APENDICE -----	91
9.1. GLOSARIO DE TERMINOS -----	91

1

INTRODUCCIÓN

Las computadoras son especialmente adecuadas para almacenar y manipular datos. Debido a que pueden ordenar y recuperar grandes volúmenes de datos con rapidez, poseen exactamente las características necesarias para administrar datos en forma eficiente.

Un sistema de administración de bases de datos (DBMS) es la herramienta que las computadoras usan para realizar el procesamiento y el almacenamiento ordenado de los datos. Una base de datos es un recipiente para colecciones relacionadas de datos. Por ejemplo, una agenda puede ser una base de datos donde se almacenan los nombres, direcciones y números telefónicos de amigos. Cada fragmento de información puede ser agregado a una base de datos y posteriormente extraerse de manera significativa. El DBMS es el programa (o colección de programas) que permite a los usuarios (y a otros programas) tener acceso y trabajar con una base de datos.

El presente trabajo está compuesto por nueve capítulos, en la segunda unidad se describen los antecedentes de la Sociedad Cooperativa de Obras por

Cooperación Ciudadana de S.C.L., como empezó a funcionar, donde y cuando se fundó y como está integrada está. Así también como la importancia de esta sociedad.

En el tercer apartado expongo el objetivo general así como los objetivos particulares que persigo.

En el cuarto capítulo llamado administración de las bases de datos, menciono qué es una base de datos, diseño de interacciones con la base de datos y el desarrollo de sistemas en el ambiente de base de datos.

Comenta el capítulo quinto la manipulación de datos con un sistema de administración; cómo el usuario usa los DBMS (Sistema de Administración de Base de Datos), los fundamentos de una base de datos, como esta integrada la base de datos, así también explico brevemente cuales son los campos que puede llevar la base de datos y para que se utilizan estos. También explico algo sobre la integración con la base de datos, algunos de los programas que se usan para base de datos. Y por ultimo hablare en esté capítulo sobre lo que esperamos tanto programadores, como diseñadores y porque no los usuarios de las computadoras.

El capítulo seis se compone del tema llamado ciclo de vida, qué es un método que se utiliza para desarrollar e implantar sistemas de información, el cuál esta compuesto de 6 actividades las cuales veremos en este trabajo.

En el capítulo siete se dan las conclusiones de este trabajo.

En el octavo tema se dan las referencias donde se consulto para la realización del presente trabajo.

Y por último en el noveno apartado se describe un pequeño pero muy útil glosario de términos.

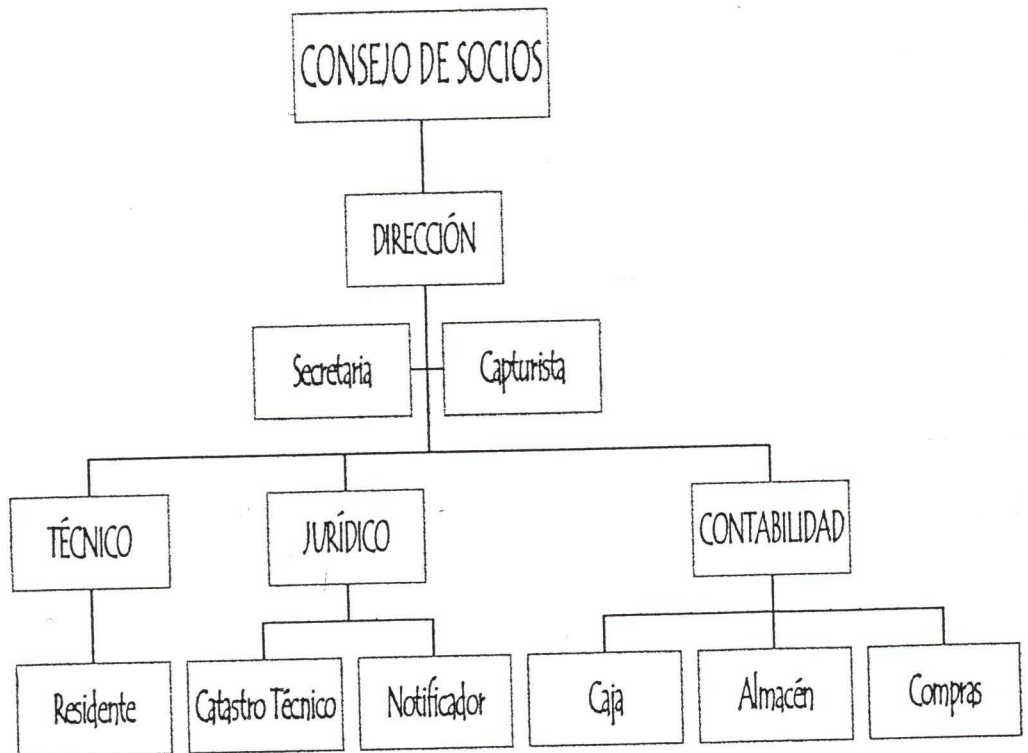
2

ANTECEDENTES

SCOOC es una Sociedad Cooperativa de Obras por Cooperación Ciudadana de S.C.L., la cual se fundó el 1 de Noviembre de 1996 con la integración de 8 socios y ha ido evolucionando con 2 oficiales, 4 albañiles y 24 peones a parte de las personas administrativas que están integradas de la siguiente manera:

ORGANIGRAMA ADMINISTRATIVO

SCOOC



La importancia que hoy en día se da a la construcción de los pavimentos dentro de cualquier población o ciudad es grande y significativa, ya que esta sirve como parámetro para determinar en forma tangible el crecimiento económico o grado de marginación en que se encuentra la ciudad en referencia.

El objetivo de SCOOC es pavimentar las calles haciendo participar a la ciudadanía organizando comités por calles, junto con los encargados del orden, transmitiéndoles seguridad y confianza con buenos servicios, creando fuentes de trabajo.

En la actualidad la información del cliente se encuentra en una hoja y es muy lento su búsqueda, el sistema es manual. No existe un sistema computarizado, que permita el ágil proceso de la información.

Albizuri Romero Miren B. Comenta en su libro "Estructuras de datos e introducción a bases de datos" que "los datos que están almacenados más o menos permanentemente en una computadora, son llamadas bases de datos. El software (programas) que permite a una o varias personas usar y/o modificar estos datos es un sistema administrador de bases de datos (DBMS)" y además que "el principal papel del DBMS es permitirle al usuario a utilizar los datos en términos abstractos y no como los almacena la computadora. De tal forma que el DBMS actúa como un intérprete permitiendo al usuario especificar que se debe hacer, con poca o nada de atención por parte del

usuario a los algoritmos detallados o a la representación de los datos usada por el sistema".

Gracias a la informática y a las bases de datos podremos facilitar este trabajo para brindarle una mejor calidad en el servicio al cliente.

Una función usual en las computadoras actuales es archivar y mantener registros de datos, para llegar a formar bibliotecas o bancos de datos. Pero a la vez que se crea uno de estos bancos de información se debe ofrecer una forma sencilla de consultar los registros, sin que el usuario tenga que escribir programas especiales. Para esto se emplean complejos sistemas, llamados manejadores de bases de datos o, más simplemente, bases de datos.

Una base de datos es un conjunto de programas que permiten crear y mantener bancos de información, y que ofrecen métodos relativamente sencillos para consultarla, guiando la búsqueda por el contenido y significación de datos, más que por su localización física. Es decir, con una base de datos bibliotecaria se puede decir, por ejemplo, un reporte de los libros que traten un tema en especial, o combinación de ellos. Como por ejemplo, un sistema manejador de base de datos aplicando a los resultados de una encuesta podría responder preguntas del tipo "¿Cuántas personas menores de treinta años, con estudios universitarios, prefieren el producto X al producto Y?".

3

OBJETIVOS

OBJETIVO GENERAL:

Desarrollar, diseñar e implantar un sistema de computo para tener un mejor control y desempeño, referente a los registros de los clientes de la Sociedad Cooperativa de Obras por Cooperación Ciudadana.

OBJETIVOS PARTICULARES:

Proporcionar el control de los clientes de acuerdo a sus necesidades con nuevos y eficientes métodos.

Obtener una óptima presentación en los informes para su uso efectivo en la toma de decisiones.

Evitar los errores humanos en cuanto al manejo de la documentación, evitando trasapeleos y ahorro de tiempo.

Realizar correctamente las operaciones sin ningún error de calculo.

4

ADMINISTRACIÓN DE BASES DE DATOS

4.1. DEFINICIÓN DE BASES DE DATOS

Es una colección integrada de datos almacenados en distintos tipos de registros, de forma que sean accesibles para múltiples aplicaciones. La interrelación de los registros se obtiene de las relaciones entre los datos, no de su lugar de almacenamiento físico.

Los registros, para distintas entidades se almacenan comúnmente en una base de datos.

Las bases de datos no eliminan la necesidad de archivos en un sistema de información. Los distintos tipos de archivos siguen siendo necesarios para capturar los detalles de los eventos y actividades de la empresa, para preparar reportes o almacenar datos que no están en la base de datos.

4.2. DISEÑO DE INTERACCIONES CON LA BASE DE DATOS

Muchos sistemas de información, ya sean implantados en sistemas de cómputo grandes o pequeños, interactúan con las bases de datos que abarcan varias aplicaciones. Dada la importancia que tienen las bases de datos en muchos sistemas, su diseño es establecido y vigilado por un administrador de bases de datos, que es una persona que tiene la responsabilidad de desarrollar y mantener la base de datos. En estos casos, el analista de sistemas no efectúa el diseño de la base de datos sino que consulta al administrador de la base para determinar las interacciones más apropiadas con la base de datos. El analista proporciona al administrador la descripción de:

1. Los datos que son necesarios de la base de datos.
2. Las acciones que tendrán efecto sobre la propia base (por ejemplo, la recuperación de datos, cambios en los valores de los datos o el ingreso de nuevos datos en la base).

A su vez, el papel del administrador de bases de datos incluye las siguientes responsabilidades:

- Evaluar la conveniencia de la solicitud del analista.
- Describir los métodos para interactuar con la base de datos.

- Asegurar que la aplicación no pueda dañar la base de datos o que la afecte de manera adversa a las necesidades de otros sistemas de información.

4.3. DESARROLLO DE SISTEMAS EN UN AMBIENTE DE BASE DE DATOS

Las bases de datos permiten compartir los datos entre distintas aplicaciones. Además de la responsabilidad de diseñar archivos, determinar sus contenidos y elegir los métodos apropiados para organizar los datos, los analistas deben diseñar los medios de interacción con las bases de datos de la organización. En la mayoría de los casos, las bases de datos ya estarán disponibles y manejadas por el personal de administración de la base de datos.

Comenzaremos por examinar los fundamentos del manejo de una base de datos, marcando las relaciones entre los datos y cómo se comparten los datos entre distintas aplicaciones.

4.3.1. RELACIONES ENTRE LOS DATOS

Cuando se diseña un sistema de información para el procesamiento de transacciones, a menudo el centro de atención es una entidad.

Cuando los analistas y usuarios van adquiriendo expresiones con el sistema de información y surgen nuevos requerimientos de la aplicación, la atención cambia: de ser capaz de recuperar un registro específico a desarrollar la capacidad de relacionar los registros sobre distintas entidades. Es probable que cambien los requerimientos cuando las firmas quieren más información para las solicitudes de procesamiento.

1. Los detalles del pedido, tales como el artículo o artículos deseados y las cantidades de cada uno, el número de solicitudes de compra del cliente y la dirección de envío.
2. Información descriptiva del cliente, como su número de cuenta, estado de crédito y gravamen de impuestos.
3. Detalles de inventario de los artículos pedidos, tales como su descripción, información de empaque.

En el ejemplo, existen varias entidades relacionadas. La toma de pedidos requiere relacionar tres entidades distintas: pedido, cliente e inventario. De eso trata el manejo de la base de datos.

1. Marcar las relaciones naturales entre los datos.
2. Compartir los datos entre entidades en todas las aplicaciones que necesiten de los detalles.

Usted puede pensar en otras relaciones entre las entidades que representan las actividades en una organización.

Por ejemplo, los departamentos están conformados por empleados; los productos tienen partes y los proyectos incluyen a los empleados. En cada uno de estos ejemplos, las entidades están relacionadas entre sí.

Es útil mostrar las relaciones en forma gráfica por medio de los diagramas de entidades - relación. Los analistas de sistemas usualmente representan una entidad por medio de un rectángulo con el nombre de la entidad dentro del rectángulo. El diamante indica la relación.

4.3.2. DESCRIPCIÓN DE LAS RELACIONES ENTRE ENTIDADES.

Las relaciones entre entidades se describe mediante su dependencia una de la otra, al igual que por el alcance de la relación.

Dependencia entre entidades.

Existen dos tipos de dependencia entre entidades. En la primera, la *dependencia existencial*, una entidad no puede existir a menos que la otra está presente; el que exista la segunda depende de la existencia de la primera.

Al eliminar los registros de una entidad de una base de datos puede ocurrir que se eliminen los registros de otra si existe una dependencia existencial. Si un cliente es eliminado ¿Se eliminarán también los registros de los pedidos hechos por ese cliente?. Esta decisión de diseño se puede hacer en

forma adecuada si el analista reconoce que existe una dependencia existencial entre las entidades.

En el otro tipo de dependencia, la *dependencia de identificación*, una entidad no puede identificarse de manera única con sus propios atributos. La identificación es posible sólo mediante las relaciones de una entidad con otras. Para identificar una entidad, se deben conocer las otras.

Alcance de la dependencia.

El alcance de la dependencia incluye dos preocupaciones interrelacionadas; la dirección de la relación y el tipo de asociación entre ellas. Ambas se pueden representar gráficamente.

Las asociaciones entre las entidades son uno a uno y uno a muchos y describen el alcance de la relación. Si es uno a uno, la aparición de una entidad quiere decir que existen una y sólo una aparición correspondiente de la otra entidad en la relación.

Es frecuente que los analistas se encuentren con que las firmas consideran a las personas u organizaciones como clientes, aun cuando no tengan en el momento un pedido pendiente. La misma firma seguramente maneja pedidos múltiples de un único cliente si éste lo hace. Esta asociación se denomina "uno a muchos". "Muchos" quiere decir 0, 1 o más apariciones de la entidad correspondiente.

Las *relaciones indefinidas*, en las que la dirección y asociación se desconocen, son inaceptables, ya que impiden el desarrollo de un modelo de datos que tenga sentido y por ende la parte de manejo de datos de un sistema de información. Así, el analista debe determinar los detalles de la relación antes de seguir adelante.

4.3.2.1. DIAGRAMAS DE ESTRUCTURAS DE DATOS.

Una vez que se han determinado las entidades y sus relaciones, podemos centrarnos en los requerimientos de datos para cada entidad. Construiremos un diagrama de estructura de datos a partir de la información obtenida, al preparar el diagrama de relación entre las entidades.

Además de los componentes básicos que ya hemos identificado en un diagrama de estructura de datos (entidades, atributos y registros), existen dos elementos adicionales esenciales:

- ◆ *Apuntadores atributos*: Enlazan dos entidades mediante la información común, usualmente un atributo llave en uno y un atributo (no llave) en el otro.
- ◆ *Apuntadores lógicos*: Identifican las relaciones entre las entidades; sirven para obtener acceso inmediato a la información en una entidad, definiendo un atributo llave en otra entidad.

Los apuntadores lógicos, que usualmente se indican en la parte inferior del diagrama de estructura de datos, son los enlaces con las demás entidades incluidas en el diagrama.

La flecha entre las entidades describe la dirección y alcance de la relación (la ausencia de un apuntador lógico indica que no existe relación entre las entidades).

4.3.3. COMPARTIR DATOS ENTRE LAS APLICACIONES.

No es usual que los analistas enfrenten a situaciones que requieran hacer la distinción entre una transacción y otra, en donde los registros se recuperan al especificar una llave de registro o bien se procesen grandes volúmenes de datos a la vez para actualizar archivos; éstas son actividades normales de procesamiento de transacciones. En vez de esto, los analistas están preocupados en las respuestas a las consultas o en la información especial que no se puede obtener utilizando llaves de registros simples.

Conviene concentrarse más en los requerimientos de distribución de datos de una organización.

Al desarrollar más sistemas y crecer su utilidad para la dirección a menudo existe la necesidad de integrar los sistemas para permitir que la información sea compartida por más de una aplicación. Análogamente, la

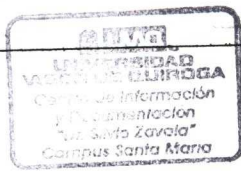
necesidad de la dirección de recibir respuestas a consultas y análisis especiales comienza a afectar el diseño de un sistema.

Redundancia e integridad

Cuando se requiere almacenamiento extra, tal duplicación innecesaria, llamada redundancia, puede reducir la integridad de la información; cuando se duplica información, hay una probabilidad alta de que los detalles no siempre coinciden. El resultado es la pérdida de integridad en los datos, problema que a veces se puede corregir mediante mejores procesamientos de oficina. Sin embargo, se puede evitar del todo disminuyendo la redundancia de los datos en los archivos.

4.3.4. EL IMPACTO DE LOS SISTEMAS DE MANEJO DE UNA BASE DE DATOS EN EL DISEÑO DE SISTEMAS.

Un sistema de manejo de una base de datos (DBMS) proporciona la flexibilidad en el almacenamiento y recuperación de datos y producción de la información. Un lenguaje de programación tal como COBOL puede incluir comandos de manejo de una base de datos (un sistema de base de datos en un lenguaje huésped) o bien, se puede desarrollar un lenguaje completamente nuevo para que soporte el manejo de la base de datos (un sistema de base de datos autocontenido).



Esquema

El uso de un sistema DBMS no elimina la necesidad de los programas de computación. El DBMS es un puente entre el programa de aplicación, el cual determina que datos son necesarios y cómo se les procesará, además del sistema operativo de la computadora, que es el responsable de colocar los datos en los dispositivos de almacenamiento. Un *esquema* define la porción de la base de datos que utilizará un programa específico. Para recuperar los datos de la base de datos:

1. El programa de aplicación determina qué datos se necesitan y comunica la necesidad al DBMS.
2. El DBMS determina que los datos solicitados realmente estén almacenados en la base de datos (aun cuando podrían estar almacenados bajo un nombre distinto, un alias). Los datos deben definirse en el subesquema (esto es posible sólo si los datos se encuentran en la base de datos).
3. El DBMS instruye al sistema operativo para localizar y recuperar los datos del lugar específico en el disco magnético (o cualquier dispositivo donde se almacenan).
4. Se da una copia de los datos al programa de aplicación para su procesamiento.

El sistema de manejo de la base de datos permite la *independencia de los datos*, lo cual significa que el programa de aplicación puede cambiar sin afectar a los datos almacenados. Cuando se utiliza un archivo maestro, si el programa se altera de forma que también se modifique el orden de los datos

almacenados o recuperados, dicho archivo debe volverse a crear y reestructurar. En cambio con la independencia de los datos, pueden ocurrir cambios en el almacenamiento o uso de un dato sin afectar a los demás. Un diccionario de datos se introduce en el sistema de manejo de la base de datos por medio del esquema y subesquema para asegurarse que los datos están definidos y descritos de forma adecuada y que la duplicidad de los nombres (alias) no produce un almacenamiento redundante de los datos o la pérdida de integridad de los datos.

Estructuras de los datos interrelacionados.

Los sistemas de manejo de datos no reemplazan las estructuras de almacenamiento tradicionales. Aun cuando los sistemas de información puedan utilizar métodos de la base de datos, éstos se seguirán almacenando mediante organizaciones secuenciales, aleatorias o indexadas. El método de organización del archivo no cambiará, ya que se basa en la forma en que opera el hardware de la computadora para almacenar y recuperar datos. Sin embargo, surgirán estructuras de datos más sofisticadas para proporcionar la flexibilidad que hemos enfatizado.

MULTILISTA:

Los sistemas de base de datos utilizan algunos de los varios métodos para estructurar lógicamente los datos. Las multilistas enlazan puntos comunes en un archivo. Una multilista es como una cadena, en donde cada eslabón es un registro que cumple con los requerimientos especificados por el usuario mediante el programa de aplicación.

Conviene hacer notar que los registros en sí no se mueven. En vez de eso, todos los registros que tienen un atributo especificado están ligados entre sí. La dirección de almacenamiento del siguiente registro en la serie se incluye en un registro para continuar con la lista, en la misma forma que los eslabones de una cadena. La única diferencia es que los eslabones están adyacentes entre sí, mientras que en la lista de la base de datos están ligados lógicamente.

El nombre de multilista se refiere a la capacidad de seguir muchas listas en una base de datos.

ARCHIVO INVERTIDO:

El otro tipo de estructura de datos que se usa comúnmente en los sistemas de manejo de una base de datos es el *archivo invertido*. Este enfoque utiliza un índice para almacenar la información acerca de la ubicación de registros con atributos particulares. En un archivo completamente invertido, existe un índice por cada tipo de datos en el conjunto de datos. Cada registro en el índice contiene las direcciones de almacenamiento de cada registro en el archivo que satisface el atributo.

Algunos datos en una base de datos probablemente nunca se utilizarán para recuperar información. Si no se indexan todos los atributos, la base de datos sólo se invierte parcialmente, lo que es una estructura más común.

Algunos sistemas de base de datos también incluyen información "numerada" en el índice: puesto que las direcciones de los registros

individuales son parte del registro del índice, el número que satisfacen un requerimiento particular se puede determinar rápidamente numerando las direcciones. Si se le pregunta al programa: ¿Cuántos clientes rentan películas?, la respuesta se puede determinar examinando simplemente el índice. Los registros de los datos en sí no necesitan ser examinados.

Las preguntas complejas que requieren de más de un atributo se pueden responder bajo cualquier de las estructuras de multilistas o de archivos invertido. Primero se realizan un listado de campos, ya sea siguiendo los enlaces en una cadena o examinando el índice apropiado. A continuación se examina la lista para el segundo atributo. Las direcciones o campos que aparecen en ambas listas satisfacen los requerimientos. Aquellos que están únicamente en una lista no cumplen.

4.3.5. MODELOS DE DATOS.

Existen tres modelos de datos que tienen uso común. El conocimiento de qué modelos de datos utilizará un DBMS determinará cómo debe estructurarse un diseño y las formas en que se representará las relaciones entre los datos.

Modelo Relacional:

El *modelo relacional* es en la actualidad el más popular en los sistemas de manejo de una base de datos, puesto que es conceptualmente sencillo y

comprensible por los profesionales de los sistemas de información y muchos otros usuarios finales; puede evolucionar ya que las relaciones entre los datos no necesitan estar predefinidas, además utiliza valores de los datos para implicar las relaciones. El modelo relacional de datos, desarrollado en 1970 por E.F. Codd, se basa en una *relación*: una tabla bidimensional. Los renglones de la tabla representan los registros y las columnas muestran los atributos de la entidad. Las *bases de datos relacionales* utilizan un modelo para mostrar cómo se relacionan lógicamente los datos de un registro.

Lo atractivo de las bases de datos relacionales es que el modelo relacional se puede comprender rápidamente.

Definición de una base de datos:

Una base de datos se debe crear antes de usarla. En una base de datos relacional, un esquema, el cual escribe la base de datos que utilizará un sistema, se comunica con el DBMS. Según el DBMS que se esté usando, las instrucciones pueden variar. La mayoría de las bases de datos relacionales utilizan un lenguaje de *definición de una base de datos*, como SQL ("Structured Query Language" Lenguaje de Consulta Estructurado).

Para crear una tabla para una relación, las instrucciones de definición de datos nombran a la relación de forma que se pueda añadir al directorio de la base de datos. Entonces, cada dato de la relación debe definirse describiendo del dato, tipo de dato y longitud.

4.3.6. ESTRUCTURACIÓN DE DATOS.

Al planear la organización de datos que va a almacenar, el analista debe prever la necesidad de acceder los datos para cumplir con requerimientos inesperados, objetivo que se puede alcanzar mediante la normalización de datos.

4.3.7. NORMALIZACIÓN.

La normalización es el proceso de simplificar la relación entre los campos de un registro. Por medio de la normalización, un conjunto de datos en un registro se reemplaza por varios registros que son más simples y predecibles y, por lo tanto, más manejables. La normalización se lleva a cabo por cuatro razones:

- ❖ Estructurar los datos en forma que se puedan representar las relaciones pertinentes entre los datos.
- ❖ Permitir la recuperación sencilla de los datos en respuesta a las solicitudes de consultar y reportar.
- ❖ Simplificar el mantenimiento de los datos actualizándolos, insertándolos y borrándolos.
- ❖ Reducir la necesidad de reestructurar o reorganizar los datos cuando surjan

Se han realizado investigaciones extensas con el fin de desarrollar métodos para normalización.

Los analistas de sistemas deben familiarizarse con los pasos de la normalización, ya que este proceso puede mejorar la calidad del diseño de una aplicación.

1. Descomponer todos los grupos de datos en registros bidimensionales.
2. Eliminar todas las relaciones en las que los datos no dependan completamente de la llave primaria del registro.
3. Eliminar todas las relaciones que contengan dependencias transitivas.

Primera forma normal.

Una de las mejores básicas que el analista puede hacer es diseñar la estructura de un registro de manera que todos los registros de un archivo tengan la misma longitud. Los registros de longitud variable crean problemas especiales, ya que el sistema debe verificar siempre en dónde se encuentran los extremos de un registro. Al fijar la longitud del registro se elimina este problema.

La primera forma normal se alcanza cuando se quitan todos los grupos de repetición, de forma que un registro tenga longitud fija. Un grupo de repetición, es decir, la aparición repetida de un dato o grupo de datos dentro de un registro, es en realidad otra relación. Por lo tanto, se quita del registro y se le considera como una parte del mismo o como una relación adicional.

La primera forma normal se alcanza cuando un registro se alcanza de longitud fija. Esto se lleva a cabo quitando el grupo de repetición y creando un archivo o relación aparte que contenga al grupo de repetición. El registro original y el nuevo se interrelacionan mediante un punto común de los datos.

Segunda forma normal.

La segunda forma normal se alcanza cuando un registro está en la primera forma normal y cada campo depende totalmente de la llave del registro. En otras palabras, el analista busca la *dependencia funcional*: un campo es funcionalmente dependiente si su valor está asociado de manera única con un campo específico. Aunque este concepto suena complejo, en realidad es muy sencillo.

Para alcanzar la segunda forma normal, cada campo del registro que no dependa de la llave primaria del registro debe quitarse para una relación aparte.

Tercera forma normal.

La tercera forma normal se alcanza cuando se quitan las dependencias transitivas de un diseño de registro.

En el manejo de datos, la dependencia transitiva es una preocupación, ya que los datos pueden perderse de manera inadvertida cuando la relación está oculta. La conversión a la tercera forma normal quita la dependencia transitiva dividiendo la relación en dos relaciones separadas.

Las dependencias transitivas requieren de un buen conocimiento de la relación entre los datos y las actividades empresariales en las que se utilizan. Pero ése es exactamente el punto. Los diseñadores del archivo y base de datos deben modelar la empresa a la que soportan. Estas decisiones no se pueden hacer en busca de satisfacciones técnicas o de almacenamiento.

Resumen de los pasos en la normalización de los datos

FORMA	PASOS
Primera forma normal	Cambiar todas las estructuras que no sean bidimensionales (es decir, grupos de repetición) en estructuras de registros bidimensionales.
Segunda forma normal	Eliminar los datos que no dependan totalmente de las llaves de registro.
Tercera forma normal	Eliminar los datos que dependan transitivamente de las llaves primarias.

La tabla resume las tres formas de normalización analizadas hasta ahora. Si la base de datos se diseña de acuerdo con los principios de normalización, la manipulación de datos, la cual se describe abajo, será más fácil (de otra forma, el analista debe trabajar con el administrador de datos para realizar los ajustes necesarios).

Manipulación de datos.

La manipulación de datos para preparar reportes responde a consultas utiliza uno de los tres operadores relacionales.

OPERADOR RELACIONAL	DESCRIPCIÓN DE LA OPERACIÓN
SELECT	Crea una nueva relación extrayendo renglones que cumplan con los criterios establecidos.
PROJECT	Crea una nueva tabla extrayendo columnas con los criterios establecidos.
JOIN	Crea una nueva relación a partir de los renglones de dos tablas que tengan atributos que cumplan los criterios establecidos.

Operaciones SELECT.

Produce una nueva tabla en respuesta a una consulta o solicitud de reporte creada a partir de los renglones de la tabla inicial que cumplan los criterios de la solicitud.

Operaciones PROJECT.

Crea una nueva tabla a partir de los datos extraídos, utilizando atributos especificados en la pregunta. En otras palabras el PROJECT escoge columnas de una relación.

Operaciones JOIN.

Creará una nueva relación combinando dos tablas existentes, eligiendo los registros que cumplan los criterios establecidos en la pregunta y removiendo después los registros duplicados. (Conceptualmente una base de datos relacional no contiene entradas duplicadas. Sin embargo, en muchas implantaciones del modelo relacional, los duplicados se eliminan sólo cuando el usuario elige la opción).

Estos operadores relacionales son la base de datos la operación en los datos. Por medio de éstos se pueden revisar las operaciones de adición, borrado y cambio.

- ◆ Adición: se añade un registro a la base de datos sin afectar los demás registros.
- ◆ Borrado: se borra el renglón de la base de datos que contiene al registro especificado.
- ◆ Cambio: el registro con la llave especificada, se localiza y los valores de los datos en el atributo secundario se cambian de acuerdo a los nuevos datos.

Los pasos necesarios para responder a la consulta o procesar las solicitudes de actualización se pueden visualizar examinando los datos. Aun cuando una solicitud particular no se haya previsto al diseñar la base de datos, ésta puede todavía ser satisfecha debido a las poderosas operaciones disponibles mediante el modelo relacional. Éste no es necesariamente el caso con los otros modelos de datos.

Modelo jerárquico.

El modelo *jerárquico* relaciona entidades por medio de una relación superior / subordinado o padre / hijo. Gráficamente, se muestra el modelo jerárquico de datos como un árbol volteando hacia arriba, en el cual el nivel más alto se conoce como la raíz. Los nodos del árbol representan las entidades.

Un modelo jerárquico de datos permite dos tipos de relación:

- *Uno a uno*: Una entidad en un nivel se relaciona con una entidad en el *siguiente* nivel.
- *Uno a muchos*: Una entidad en un nivel se relaciona con una, muchas o ninguna entidad del *siguiente* nivel.

El analista de sistemas se ve afectado por las decisiones hechas al diseñar una base de datos jerárquica. Durante el diseño, el administrador de la base de datos, quien es responsable del diseño, determina las entidades a incluir en la base de datos y la relación que existirá entre ellas. Los nodos representan ocurrencias de registros que contienen los datos tal como determina el administrador de los datos.

El diseño de una base de datos jerárquica afectará la facilidad de acceso a los datos (las notas de acceso no necesitan hacerse al utilizar bases de datos relacionales).

El analista de sistemas debe trabajar con las restricciones de diseño que surjan. Por ejemplo, el modelo de datos estipula que los artículos son accesibles sólo mediante la aparición de un registro de pedido. Esta relación indica que se debe buscar toda la lista de artículos de todos los pedidos para preparar un reporte de la venta histórica de los productos.

Aparecen efectos colaterales indeseables bajo ciertos diseños de una base de datos. Las bases de datos jerárquicas involucran anomalías con respecto a lo siguiente:

- ❖ Inserción de registros: Un registro dependiente no se puede añadir a la base de datos sin un padre.
- ❖ Borrado de registro: Al borrar un padre de la base de datos también se borran todos sus descendientes.

Si estas situaciones ocurren con cierta probabilidad en el marco de una aplicación particular, es necesario establecer múltiples copias de los registros e incluso bases de datos múltiples para evitar el problema.

Modelo de red.

El modelo de red es análogo al modelo jerárquico, excepto que una entidad puede tener más de un padre. Los miembros pueden pertenecer a más de una relación (es decir, tienen más de poseedor).

En las bases de datos de tipo red, así como en las jerárquicas, se deben establecer las relaciones entre las entidades al mismo tiempo que se establece el modelo de los datos y se crea la base de datos (en contraste con el modelo relacional, el cual no requiere rutas de acceso predefinidas o relacionales entre las entidades). El analista de sistemas debe ajustarse a esos detalles cuando desarrolla aplicaciones que capturan o recuperan datos durante el procesamiento.

Las bases de datos jerárquicas y de red son conceptualmente sencillas y parecen no ser complicadas a primera vista. Sin embargo, en un ambiente de una base de datos grande, pueden evolucionar rápidamente hacia una telaraña complicada de interrelaciones que son difíciles de manejar al evolucionar la base de datos con el uso.

Aparecen anomalías similares a las del modelo jerárquico de datos.

Existe un lado positivo de los modelos de datos jerárquicos y de red, el cual deben observar los analistas de sistemas. Supongamos que se pueden predefinir las rutas de acceso y relaciones entre las entidades (cuando el esquema se desarrolla y se crea la base de datos). Sin los requerimientos de acceso y recuperación del desarrollo de la aplicación se adecuan a las rutas de acceso predefinidas, será más rápido el procesamiento de consultas, actualizaciones y adiciones a la base de datos que cuando se utilizan bases de datos relacionales.

Los analistas de sistemas también reconocen que existen muchísimas bases de datos basadas en los modelos jerárquicos y de red instaladas en la actualidad en la comunidad empresarial. Si estas bases de datos están cumpliendo los requerimientos operacionales, es improbable que las organizaciones las reemplacen. Así, el desarrollo de las aplicaciones de sistemas de información que tomen en cuenta las oportunidades y restricciones que ofrecen es una necesidad.

Diseño en un ambiente de base de datos.

Aunque los analistas no diseñen las bases de datos, a menudo desarrollan sistemas que implican DBMS y bases de datos. Esta sección analiza las consideraciones que debe tomar el analista en estos casos.

En la mayoría de las organizaciones donde las bases de datos se utilizan y manejan bien, el personal responsable de la base de datos vigila el diseño y desarrollo de la propia base de datos, define el esquema, mantiene el diccionario de datos y busca estándares para los datos (tales como nombre, tipo, longitud y uso).

El analista de sistemas debe aun determinar los requerimientos de la información y las especificaciones de procesamiento de programa, las que a su vez se traducen en el contenido del subesquema. El subesquema, analizado previamente, es la definición lógica de los datos para la base de datos que utiliza el programa. Consta de nombre y descripción de los datos y es un subconjunto del esquema. Para cada base de datos existe un único esquema,

pero pueden existir muchos subesquemas. Cada aplicación de sistemas de información que utilice la base de datos puede tener un subesquema diferente.

El analista de sistemas es el responsable de determinar cómo se ligarán los datos para cumplir con los requerimientos del usuario. Los análisis de requerimientos realizados con cuidado permiten a un analista prever consultas complejas y proporcionar los métodos para responder a ellas.

Las estructuras de registro son determinadas por el analista de sistemas. El conocimiento que ciertos datos son necesarios para proporcionar información importante al usuario sigue siendo responsabilidad del analista. Sin embargo, puesto que en un ambiente de base de datos las estructuras de riesgo son lógicas (refiriéndose a la forma en que el programa verá los datos), pueden ser distintas las estructuras físicas reales (la forma en que el sistema almacena los datos). Como hemos analizado, las diferencias entre los DBMS ayudará a hacer la transición necesaria de reformateo de los datos para dar al programa lo que espera. La estructura de los datos debe seguir los criterios de longitud y tipo especificados en el diccionario de datos.

Si un sistema utiliza una base de datos existente, el analista debe completar los siguientes pasos:

1. Familiarizarse con el esquema de la base de datos.
2. Revisar los estándares y especificaciones de los datos en el diccionario de datos.

3. Determinar los requerimientos lógicos de los datos y, trabajando con el personal de administración de los datos, desarrollar un subesquema que se adecue al esquema a la vez proporcione los datos requeridos por el sistema de información.
4. Identificar y diseñar los archivos maestros o de transacción necesarios para el sistema de información, pero que no forman parte de la base de datos existente.
5. Determinar los esquemas apropiados de identificación, código y validación, así como los procesamientos adecuados de procesamiento para todas las entidades, ya sea que los datos o en archivos separados.
6. Considerar las preocupaciones de diseño, de métodos de entrada y salida, los cuales no cambiarán.

Si los analistas están trabajando con un sistema que implica el desarrollo de una nueva base de datos, todos los pasos anteriores son necesarios, además, se necesitarán una interacción cercana con el personal de manejo de la base de datos para establecer el esquema, definiciones y diccionario de datos.

El DBMS no reemplaza los principios de diseño de sistemas que hemos analizado. En vez de eso, proporciona al analista una herramienta adicional para usarla en el desarrollo de sistema que proporcionen información esencial a la dirección. Las preguntas acerca de los requerimientos del usuario guían cualquier decisión con respecto al diseño de sistemas o uso de un software de base de datos. El objetivo del diseño es proporcionar la información correcta

en el momento oportuno. Así, el uso de un DBMS no es un objeto, sino un medio para lograr un fin.

5

MANIPULACIÓN DE DATOS CON UN SISTEMA DE ADMINISTRACIÓN

Las computadoras son especialmente adecuadas para almacenar y manipular datos. Debido a que pueden ordenar y recuperar grandes volúmenes de datos con rapidez, poseen exactamente las características necesarias para administrar datos en forma eficiente.



Un sistema de administración de base de datos (DBMS, database - management system) es la herramienta que las computadoras usan para realizar el procesamiento y almacenamiento ordenado de los datos. Una base de datos es un recipiente para colecciones relacionadas de datos. Cada fragmento de información puede ser agregado a una base de datos y posteriormente extraerse de maneras significativas. El DBMS es el programa (o colección de programas) que permite a los usuarios (y a otros programas) tener acceso y trabajar con una base de datos.

Los programas para bases de datos destinados a computadoras personales vienen en muchas formas, tamaños y variaciones. Algunos de estos programas para PC son desarrollados por las mismas compañías que hacen hojas de cálculo, procesadores de palabras y demás software de gran demanda. Entre éstas se incluyen dBase IV y Paradox de Borland International; Access y Foxpro de Microsoft; Q & A de Symantec; Lotus Approach y FileMaker Pro de Claris. Para sistemas más grandes y para computadoras con sistema UNIX y OS/2, algunos de los DBMS disponibles son Oracle, Ingres, Informix y OS/2 Database Manager.

5.1 CÓMO SE USAN LOS DBMS

Los DBMS son algunas de las aplicaciones más comunes usadas en las computadoras. A partir de que las primeras computadoras comerciales

estuvieron disponibles, los DBMS han constituido uno de los usos primordiales de las computadoras en el mundo de los negocios. Las compañías de todas las industrias dependen en gran manera de los DBMS. Los miles de bancos nacionales y locales que se integran a redes de banca mundiales, compañías manufactureras y empresas e instituciones en campos del cuidado de la salud, por nombrar sólo algunas, almacenan volúmenes inconcebibles de transacciones e información todos los días.

Una compañía de venta de menudeo puede usar DBMS para registrar cada transacción realizada en cualquiera de las terminales de venta que haya en la tienda. Después de cada venta, se ejecuta un programa que actualiza la base de datos que contiene la información de ventas, el mismo programa actualiza después la base de datos del inventario para eliminar del stock en almacén los artículos vendidos. Una vez que estos registros están almacenados en las bases de datos de la compañía, se pueden llamar nuevamente en cualquier momento.

Aparte de estos sistemas especializados, existen cientos de aplicaciones de administración de bases de datos de propósito general, que cada compañía e institución puede utilizar, incluidos los sistemas de contabilidad, administración de personal y nóminas.

Las computadoras personales han traído la administración de las bases de datos a los escritorios de las personas, tanto en el hogar como en la oficina. Aunque el individuo promedio generalmente no necesita un sistema de

seguimiento de inventario en casa, son muchos los usos personales para los DBMS. Mucha gente con computadoras en casa usan productos de software comercial para administrar las compras domésticas, mantener contacto con los amigos y de negocios, e incluso para preparar menús y listas de compras y para categorizar y ordenar cupones de mercancías.

5.1.1 LOS DBMS

Un DBMS es un programa o colección de programas que almacena los datos en forma tal que se permita el acceso a los mismos en cualquier momento. Puesto que los archivos de bases de datos pueden ser muy grandes (quizá incluso de muchos gigabytes en sistemas grandes), poder invocar los datos con rapidez no es un asunto trivial. Una base de datos puede proporcionar acceso rápido, en especial cuando se corre en hardware potente.

De hecho, una de las mejores razones para usar una base de datos no es sólo la velocidad para recuperar los datos, sino también, la fácil y rápida manipulación de los mismos. Actualmente, incluso en sistemas caseros, la posibilidad de cambiar la organización de los datos o de editarlos en forma individual es lo que hace tan atractivo el uso del DBMS electrónico.

Además de proporcionar acceso veloz y la posibilidad de manipular los datos, un DBMS también proporciona un medio para enlazar (o relacionar) datos contenidos en bases de datos separadas. Por ejemplo, puede analizar

rápidamente los tipos de productos que un consumidor compra más a menudo si relaciona la información del consumidor y la información acerca de los pedidos.

5.1.2. PROGRAMAS Y LOS DBMS

Al igual que un sistema operativo, un DBMS puede proporcionar servicios tanto a usuarios como a otros programas. A menudo, cuando el usuario piensa que está usando directamente un DBMS, lo que realmente hace es usar un programa que proporciona una interfaz de usuario que trabaja con el DBMS para proporcionarle acceso a las bases de datos.

En el corazón de un DBMS se encuentra un programa denominado motor de la base de datos. El motor de la base de datos es parecido al núcleo del sistema operativo, ya que ambos pueden tener acceso y manipular a un nivel bajo: hardware, en el caso de un sistema operativo, y archivos conteniendo datos, en el caso de un motor de base de datos. El motor de la base de datos recibe instrucciones de otros programas. Las instrucciones que el motor recibe pueden incluir "abre un archivo", "localiza al cliente X", "agrega un nuevo cliente" y cosas por el estilo.

5.1.3. BASES DE DATOS INTERACTIVAS.

La mayoría de los DBMS para PC incluyen poderosos programas de acceso a los datos en una base de datos. En algunos de ellos, el programa es el propio DBMS y es probable que no puedas hacer la distinción entre el motor de la base de datos y la interfaz. En Access de Microsoft, se representa una interfaz Windows estándar que incluye una opción de menú File. Al usar el menú File, tu puedes abrir las bases de datos y cambiar, borrar, eliminar o agregar información nueva.

5.2. FUNDAMENTOS DE BASES DE DATOS

En esta sección, examinaremos las tuercas y tornillos de los DBMS, para ver qué los hace funcionar. En principio, observaremos la estructura de las tablas de base de datos, incluidos los campos para almacenar diferentes tipos de datos, después analizaremos las claves y los índices (las herramientas que el DBMS usa para ordenar datos y recuperarlos rápidamente). Finalmente, examinaremos los formatos (nuestra ventana a la base de datos) y se describirán algunas de las técnicas y métodos para crear relaciones entre las tablas.

5.2.1 ESTRUCTURA DE LAS BASES DE DATOS

La siguiente figura muestra una tabla compuesta de columnas, filas o renglones. Es una base de datos con información de clientes. Un conjunto particular de datos como éste se denomina *tabla*: es un arreglo de columnas y filas. Cada cliente en la tabla aparece en una fila separada. A las columnas de la tabla se les llaman campos, y a las filas se les llaman registros. Cuando agregas el nombre de un cliente a una base de datos de clientes, se agrega una nueva fila que contiene un espacio para cada campo destinado al registro nuevo. De igual manera cuando procesas una venta y generas una factura se agrega un registro a una base de datos que contenga facturas.

RFC	NOMBRE	TELEFONO	CODIGO POSTAL
PRSK290976	Karla Pamela Pureco Sánchez	15-45-27	58020
NVLJ130175	Javier Nieves López	15-43-34	58220
ARGJ250276	Julieta I. Arriaga García	14-79-97	58280
PTHL120776	Lucía Pantoja Hurtado	15-57-98	58280
DTGSO11175	Sofía Daut García	15-89-22	58280
AGGK160984	Kenia Alejandra Aguilar García	26-27-54	58230

Campos:

Los campos de una tabla separan los tipos de información contenidos en ella. Por ejemplo, cada registro en la tabla de clientes tiene un nombre, dirección y número telefónico, y cada campo existe en cada uno de los registros. La secuencia de los campos en una tabla de base de datos está definida estrictamente para cada registro. Por ejemplo, el campo de número telefónico debe contener un número telefónico para cada registro (no puede contener un número telefónico en algunas filas y un número de seguro social en otras).

Los campos de una base de tabla de datos son definidos por la persona que crea la tabla. Los diferentes productos de DBMS ofrecen una variedad de tipos de campo. Los tipos de campo más empleados comúnmente son los campos de texto, los campos de fecha y los campos numéricos, aunque son necesarios otros tipos especiales en algunas situaciones.

Campos de texto:

Un *campo de texto* almacena una cadena o serie de caracteres alfanuméricos. Un campo de texto puede contener el nombre de una persona, el de una compañía, una dirección o cualquiera otra dirección importante en texto. Un campo de texto también puede contener números pero son tratados como si fuesen una serie de dígitos y no como el número que representan. Para datos numéricos que no vaya a emplear en cálculos, un campo de texto es lo adecuado (por ejemplo, el caso de un código postal). No sería adecuado que usara un código postal en cálculos, y como campo de

texto podría ordenar alfabéticamente las direcciones de otros países que en ocasiones contienen letras. Por otra parte, en los casos de los códigos postales de los Estados Unidos, por ejemplo, 07760 se mantiene el cero inicial en su sitio. Un campo numérico automáticamente eliminaría el 0 inicial, creándose confusiones.

Campos numéricos y de tipo de moneda:

Los *campos numéricos* almacenan números. En la mayoría de los programas puede seleccionar un formato de exhibición para los números. El número real en el campo no contiene ningún formato, pero cuando el programa exhibe dicho número puede agregar un separador de coma entre los miles y millones desplegar o no la precisión a la derecha del punto decimal incluir otros caracteres especiales como el signo del peso (\$).

Algunos programas de base de datos proporcionan más de un tipo de campo numérico. Un campo numérico puede estar limitado en cuanto al rango de valores que pueda contener. Por ejemplo, el *campo de número corto* en Paradox solamente puede contener números enteros entre -32,767 y +32,768, pues utiliza solo dos bytes para almacenar el número. Si sabe que su base de datos usará, este campo únicamente para números enteros en dicho rango, puede ahorrar espacio de almacenamiento y una pequeña cantidad de tiempo de procesamiento si utiliza un campo de números cortos. En caso de *números grandes y números que necesitan punto decimal flotante*, podría usar un *campo de número regular*, que usa más bytes de espacio de almacenamiento.

Un *campo de tipo monetario* es un campo de numeración decimal con el formato de exhibición establecido por el software para representar dinero. Un campo de tipo monetario exhibe sus valores con separadores de coma, dos lugares decimales de precisión para los centavos y, en ocasiones, un signo de peso. Internamente, los programas de DBMS tratan a los datos en los campos de tipo de moneda como si fuesen un tipo especial de números. A menudo, el número real almacenado en el campo no es un número de tipo flotante (un número con punto decimal). Los números de punto flotante ocupan más espacio de almacenamiento del que es necesario para los números que nunca tendrán una precisión de más de dos lugares decimales. En general, el DBMS convierte el peso y los centavos a centavos al multiplicar el valor por 100. Se necesita menos espacio para almacenar un número entero que sea 100 veces más grande, que el necesario para almacenar números con punto decimal flotante.

Campos de fecha y hora:

Los *campos de fecha* y los *campos de hora* son campos especializados. Al igual que las fechas y horas almacenados en las celdas de una hoja de cálculo, las fechas y horas en una base de datos se almacenan internamente como un número, pero se exhiben ya sea como una fecha o una hora. Cuando introduces una fecha o una hora en un campo de fecha / hora, el DBMS acepta la entrada en el formato de fecha o de hora, pero convierte el dato en un número antes de almacenarlo en la base de datos. De esta manera, los datos ocupan menos espacio de disco y puedes emplear fácilmente las fechas y las horas en cálculo.

Además de convertir las fechas a números, para facilitar el almacenamiento y cómputo, la mayoría de los productos de DBMS proporcionan verificación automática de errores de fecha y horas. Por ejemplo, cuando introduce una fecha en un campo de este tipo, el programa de DBMS la verificará para asegurar que se trate de una fecha válida. La mayoría de los sistemas toman en cuenta los años bisiestos, asegurando que cada fecha que se capture sea válida.

Campos lógicos:

Un *campo lógico* (llamado en ocasiones sí / no) es un campo que puede almacenar uno de sólo dos valores. Los campos lógicos se pueden usar para cualquier tipo de datos donde sólo haya dos valores posibles, aunque las descripciones que de para las opciones sean ilimitadas (sí o no, verdadero o falso, activado o desactivado, menudeo o mayoreo, etc.).

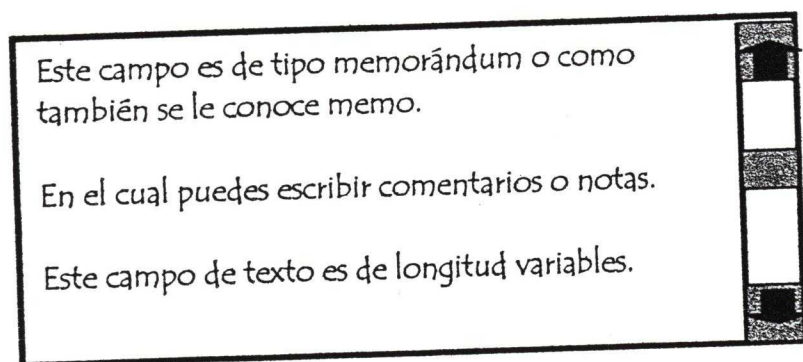
Dado que un campo lógico puede contener sólo dos valores posibles, únicamente ocupa un bit (1/8 de byte) para almacenar un valor lógico.

Campo de memorándum (memo):

Un *campo de memorándum (memo)* es un campo especial que puede contener información de longitud variable. Por ejemplo, en una agenda, emplearía campos de texto para los nombres, las direcciones y los números telefónicos de las personas. Los campos de texto normalmente tienen una longitud máxima ya fija y, la mayoría de las veces, una longitud máxima no representa un problema. Sin embargo, digamos que quiera agregar un campo

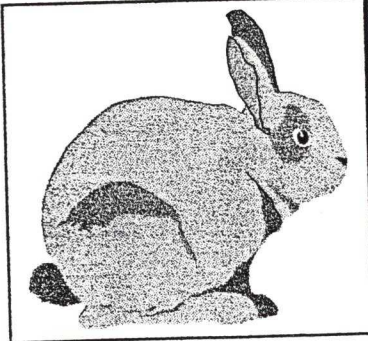
para comentarios o notas. No es posible saber de antemano cuán grande se debe hacer un campo de texto para este tipo de datos. En muchos registros, quizá no introduzca nota alguna, pero en otros, tal vez quiera guardar algunas palabras o una pequeña frase. E incluso habrá otras donde quiera poner muchas más notas y comentarios.

Un campo de memorándum resuelve este problema. Está diseñado justo para este propósito: almacenar información de longitud variable. Un campo de memorándum puede lucir como un campo de texto regular que se extiende hasta la izquierda o la derecha, o como un cuadro con desplazamiento vertical. Algunos DBMS almacenan campos de memorándum internamente, dentro de una tabla; otros crean un archivo separado para la captura de los datos de un campo de memorándum y crean y mantienen un enlace con ese archivo.



Campos binarios:

Un *campo binario* es un campo especial para almacenar objetos binarios o BLOB (Binary Large Object; Objeto binario grande). Un BLOB puede ser un archivo gráfico de imagen, por ejemplo, un recorte artístico, una fotografía, una imagen de pantalla u otra muestra de arte gráfico o texto con formato. Un BLOB también puede ser un archivo de bandas sonoras y videos, objetos OLE, como gráficos u hojas de trabajo creadas con una hoja de cálculo o procesador de palabras, e incluso archivos binarios externos, como una hoja de cálculo o un documento de procesador de palabras.

Nombre:	<input type="text"/>
Categoría:	<input type="text"/>
Macho o Hembra:	<input type="text"/>
Edad:	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
	

Campos calculados:

Algunos DBMS permiten la creación de campos especiales para realizar cálculos. Un cálculo puede ser la suma de otros campos numéricos o una fórmula para calcular el impuesto en un subtotal. Los campos están asociados con la pantalla, o *forma* que un programa exhibe, más que con la tabla de la base de datos en sí.

No todo DBMS proporciona campos calculados. En aquellos que no, los cálculos son realizados por el programa que crea y actualiza la pantalla o que genera los informes.

Campos de contador:

Un *campo de contador* contiene un valor numérico único que el DBMS asigna para cada registro. Cuando introduces un registro en una tabla vacía y nueva, con un campo de contador, el campo de contador para dicho registro se establecerá en 1. Al siguiente registro se le asigna 2 y así sucesivamente. Un campo de contador en una tabla asegura que cada registro tenga un valor completamente único en dicho campo.

5.2.2. ORDENAMIENTO DE LOS DATOS

Una de las características más poderosas de los DBMS es su capacidad para ordenar la información que contienen, ya sea para emitir un informe

impreso o para que la revise en el monitor de la computadora. Los DBMS usan claves o índices para generar datos ordenados.

Clave principal:

Una *clave principal* define un orden preestablecido para una tabla. Éste es el orden en que se despliegan los registros de una tabla en la pantalla o se imprimen en un informe, en caso de no especificarse otro orden. Una clave principal generalmente está asociada con un campo individual en una tabla (por ejemplo, un número de identificación de un cliente) y los valores que capture en dicho campo deben ser únicos; es decir, diferentes para cada registro en la tabla. Al definir un campo con los números de cliente como la clave principal, se evitará que los usuarios inadvertidamente capturen datos duplicados (dos clientes con el mismo número de identificación).

La clave principal de una tabla se puede componer de más de un campo. A este tipo de clave principal se le llama *clave compuesta*. Por ejemplo, la clave principal puede contener tres componentes: apellido, nombre e inicial del segundo nombre. Con este arreglo, el orden preestablecido en la tabla será de acuerdo con el apellido en primer lugar, y por cada apellido idéntico, los registros se ordenarán de acuerdo con el nombre y así sucesivamente. No sería adecuado que definiera una clave principal sólo para un apellido o para una combinación de apellido - nombre, pues al hacerlo eliminaría la posibilidad de capturar dos registros con el mismo nombre.

En la mayoría de las bases de datos, cuando introduce un registro nuevo, éste se agrega a la parte final del archivo que contiene los datos de la tabla. Cualquier cantidad de claves de ordenamiento o de cambio no afectará nunca el orden físico de los registros en este archivo, puesto que la base de datos mantiene la información de la clave separada de los datos. La clave es en sí una presentación en miniatura de la tabla de datos, pero contiene sólo el campo de la clave y un número que corresponde al número de registro físico en la tabla de datos. De cualquier manera, la tabla clave está ordenada. Para encontrar un registro mediante un valor de clave específico, el DBMS puede encontrar rápidamente la información requerida en la tabla de clave (ya que está ordenada), hace una nota del número de registro físico y después regresar al archivo de datos y moverse directamente hacia los datos del registro, usando el número del registro.

Índices:

Un índice es esencialmente lo mismo que una clave. En Access de Microsoft, la clave principal es especial. Se requiere de una para cada tabla para asegurar que los registros sean únicos y la clave especifica el orden preestablecido de los registros. Sin embargo, también puede asegurar otras claves llamadas *índices*, para otros tipos de ordenamiento que quiera en reportes o para agilizar las búsquedas.

Por ejemplo, en una tabla con datos de clientes, la clave principal es la identificación del cliente. Hacemos que el campo de identificación del cliente sea la clave principal porque se quiere que el orden esté determinado por dicho

campo. Sin embargo, en ocasiones tal vez desee imprimir una lista de los clientes, que esté ordenada por nombre de la compañía, ciudad o por algún otro campo. Para lograr esto, puede definir un índice para el campo o los campos que quiera ordenar.

La definición de índices también proporciona otro beneficio. Puede lograr que en la base de datos se busquen registros específicos mucho más rápido. Por ejemplo, si busca con mucha frecuencia valores en determinado campo (como la búsqueda de un apellido en un campo), si el campo no está definido como un índice, la localización de registros puede tomar desde algunos segundos hasta un par de minutos, dependiendo del tamaño de la tabla y del número de registros en ella. La búsqueda de un valor específico en un campo clave (o índice) es mucho más rápida, de hecho, es casi de inmediata pues el DBMS usa la información de clave preordenada para localizar valores de clave rápidamente y para determinar la ubicación física de los datos clave en el archivo de datos.

5.2.3. FORMAS

Aunque anteriormente se vio una tabla de base de datos muy parecida a una hoja de cálculo, la mayoría de las veces no es conveniente trabajar con una base de datos en esa forma. Los programas de base de datos te proporcionan herramientas para construir *formas*; es decir, pantallas personalizadas para exhibición y captura de datos. Una forma puede estar

enlazada a una tabla individual o a tablas múltiples con relaciones que el usuario defina.

La forma preestablecida:

La mayoría de los programas de bases de datos proporcionan un método rápido para crear una forma rudimentaria con la cual capturar los datos automáticamente.

Las formas "preestablecidas" no son muy llamativas, pero su creación es fácil y rápida. También puede crear formas completamente personalizadas; pero es más fácil tener una forma preestablecida y después personalizarla como mejor le parezca.

Campos y herramientas de control:

Con las bases de datos gráficas, cuando pone un campo sobre una forma, selecciona una herramienta para captura de datos adecuada para el campo en cuestión. Una *herramienta para captura de datos* es el mecanismo que el usuario utilizará para capturar datos en el campo. Por ejemplo, en una forma de factura, los campos para el nombre y la dirección del cliente deben ser cajas de texto. Para esto no se necesitan controles fabulosos, pues la información de cada cliente nuevo se debe capturar manualmente. En estos campos, sin embargo, los controles deben ser más apropiados. Una *caja de lista* es un control de campo que proporcionan algunas opciones del mismo, de las cuales el usuario puede elegir una.

La *caja de verificación*, es el apropiado para los campos lógicos. Muestra una marca de selección cuando se hace clic sobre ella.

Los *botones de radio* son convenientes cuando sólo una opción es posible de entre varias.

Una *caja de menú en cortina* se parece a un campo de edición, pero tiene una flecha, desciende un cuadro con opciones, debajo del campo. Al seleccionar una opción se cierra la caja y se captura dicha opción en el campo.

Una *caja de combinación* combina un campo de edición y una caja de menú de cortina. En una caja de combinación puede hacer clic sobre la flecha y desplegar una lista de valores comunes para el campo o escribir un nuevo valor en la caja de edición.

Verificación de una entrada con enmascaramiento:

Además de seleccionar un control para un campo y facilitar la factura de datos, también puede hacer una forma que ejecute validaciones o conversiones en los datos capturados con una especificación de enmascaramiento (con mascarillas). En algunos sistemas, las mascarillas están asociadas con formas para las capturas de datos y usted las especifica mientras crea la forma; en otros, las mascarillas están asociadas con los campos de la tabla y usted las especifica al tiempo que defina los campos durante la creación de la tabla.

Un enmascaramiento puede tener una función sencilla por ejemplo, convertir los caracteres escritos a mayúsculas o minúsculas. Usar un enmascaramiento para hacerse cargo de las conversiones entre mayúsculas y minúsculas pueden facilitar enormemente la captura de datos y asegurar que estos se capturan de manera uniforme (cosas que después es vital, cuando se realizan las búsquedas de datos específicos).

Un enmascaramiento también puede exhibir los contenidos de un campo en un formato determinado.

La mascarilla también se puede usar para propósitos más complejos; por ejemplo, para hacer que la posición de cierto carácter sea una letra o un número, o para hacer que la longitud de un dato capturado sea de una cantidad específica de caracteres.

La sintaxis de las especificaciones de enmascaramiento, así como las posibilidades que tiene con él, varían grandemente de un producto a otro. De hecho, incluso el nombre de esta característica varia. En Access de Microsoft una mascarilla recibe el nombre de *formato* ("format") de campo. En Paradox recibe el nombre de *láminas* ("pictures").

Paradox para Windows tiene un método especialmente amigable para el desarrollo de la lámina para un campo, su verificación y uso.

Maquillaje de una forma:

La mayoría de los programas de base de datos para ambientes gráficos le permite personalizar las formas tanto como lo desees. Además de poder asociar controles con campos, también puede especificar texto y colores de plano de fondo, o tipos de letra y atributos, e incluso imágenes gráficas como logotipos o diseños en una forma.

El grado en que se puede maquillar una forma depende del medio en que se opere el sistema utilizado. Los productos para el ambiente Windows o para Macintosh ofrecen muchas posibilidades. En otros DBMS no gráficos, estas limitado a gráficos basados en un texto, que incluyen cuadros de uno y dos contornos, y otros caracteres disponibles en el conjunto de caracteres ASCII.

5.2.4. ENLACE DE TABLAS DE BASE DE DATOS

Hasta el momento se han analizado las bases de datos, tablas campos y las formas que dan acceso a una tabla y a sus campos. Por otra parte, hemos notado que en algunas bases de datos para computadoras personales, el término bases de datos hace referencia a un sólo archivo de computadora con una tabla de información dentro. La definición más extensa de una base de datos no obstante, abarca más de una tabla relacionada de información dentro de una base de datos. Por ejemplo, en un sistema de contabilidad, necesita un sitio para almacenar los nombres, las direcciones y otra información

Cuando examina una tabla individual de datos, tal vez le impresione la utilidad de tener la información en una base de datos donde puede ordenar fácilmente en diferentes campos para proporcionar otras perspectivas de los datos. Sin embargo, la potencia real de un DBMS deriva de la capacidad para crear enlaces entre las tablas. Un DBMS capaz de hacer esto recibe el nombre de *sistema de administración de bases de datos relacionales (RDBMS*, por sus siglas en inglés).

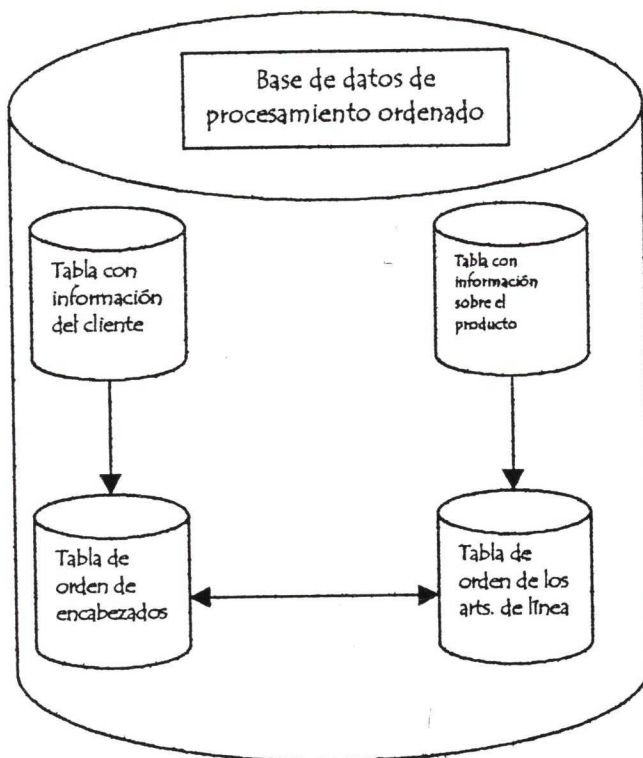
Las bases de datos no relacionales se denominan *bases de datos para archivos planos*. Las bases de datos para archivos planos son ideales para muchas aplicaciones, especialmente listas; por ejemplo, una lista de clientes o de contactos. En años recientes, las bases de datos para archivos planos se han vuelto cada vez más populares debido a su simplicidad. Mucha gente compra y usa bases de datos para archivos planos en sus sistemas caseros y en pequeños negocios. Q & A de Symantec y Lotus Approach son dos ejemplos populares de DBMS para archivos planos.

Sin embargo, las bases de datos relacionales proporcionan mucho más poder y flexibilidad y, como resultado, por lo general son la opción para sistemas de bases de datos muy grandes.

Relación de uno a muchos:

Una factura impresa tiene un área de encabezado que contiene información del consumidor, número de orden, fecha de factura, fecha de envío e información que se relaciona con la venta en general. Por otra parte

pertinente para cada cliente. De igual manera necesita almacenar información acerca de los vendedores (compañías o gentes a quienes les compras productos). Además, necesita almacenar información acerca de precios, costos de envío, etc.; de los productos de su compañía. Dado que en cada una de estas clases de información es diferente, necesita una tabla separada para cada una de ellas. Pero no obstante que los sistemas usen tablas múltiples para almacenar cada tipo de información por separado, los datos que cada tabla contiene están relacionados. Todas estas tablas, consideradas como un todo, constituyen la base de datos.



Una base de datos es una colección de tablas de datos relacionados. Las formas y los informes pueden enlazar las tablas de manera útiles. En un sistema de procesamiento ordenado, las tablas principales son las tablas de orden de encabezado y de orden de detalles (tabla de artículos de línea), sin embargo, distintas tablas suplementarias también pueden proporcionar alguna otra información importante.

las facturas tienen un área de detalle donde se enumeran, por línea de producto, los artículos que el cliente haya comprado.

Pensemos por un momento en la estructura de la base de datos que necesitaría para almacenar información de la factura. Si diseño una tabla individual para estas dos áreas, tendrá que decidir cuantos artículos de línea se podrán incluir en cada factura. Esto ocasionaría desperdicio del espacio del disco y haría que el procesamiento de la factura fuese lento.

La solución es crear dos tablas separadas y definir una relación entre ambas. Este es el tipo más común de relación entre la tabla de base de datos y se denominan *relación de uno a muchos* o *relación entre encabezados y detalles*. Con este método, la información del encabezado se guarda en una tabla de encabezado y la información de detalle se guarda en una tabla de detalle. Las dos tablas están enlazadas por un campo común.

En la mayoría de los sistemas de bases de datos, la creación de enlace es un procedimiento sencillo.

Para crear una tabla de dos formas en Access de Microsoft, primero diseña dos formas independientes una de otra: una forma principal asociada con la tabla principal (encabezado) y otra forma asociada con la tabla de detalle (número telefónico).

La forma de detalle (que se denomina subforma), se diseña para mostrar registros múltiples simultáneamente en un arreglo tabular. Una vez que ambas tablas han sido creadas, puede simplemente arrastrar la subforma con el ratón y dejarla en la forma principal. Access establecerá el enlace en forma automática.

Enlace de tablas para búsquedas:

Otra manera en que una forma puede usar tablas múltiples consiste en validar entradas y simplificar la captura de datos con campos de búsqueda.

La búsqueda también se puede usar con menú de cortina y cajas de combinación. De esta manera, el enlace no solo validará una entrada en el campo sino que también ayudaría al usuario a realizar una selección. Si se presiona la flecha así abajo en el control de la caja puedes desplegar las opciones en la tabla de referencia cruzada y realizar una selección con el ratón.

5.3. INTEGRACIÓN CON LA BASE DE DATOS

La mayor parte del tiempo que invierte en el uso de una base de datos se consume en el trabajo interactivo con las formas de la base de datos, para actualizar registros existentes y crear registros nuevos.

5.3.1. NAVEGACIÓN, VISTA Y EDICIÓN DE REGISTROS.

Examinaremos los comandos de movimiento más simples: ir directamente al primero o último registro en una tabla y trasladarse entre ellos para revisar registros, de uno en uno. También investigaremos la mecánica en el uso de una forma para agregar registros nuevos y actualizar los existentes.

Desplazamiento entre registros:

Una vez que haya cargado una tabla o establecido una forma, el programa de la base le da sus proporciones mecanismos para moverse dentro de la tabla. Los botones de la barra representan los comandos para moverse entre los registros. De izquierda a derecha significan:

- Ir al primer registro en la tabla.
- Ir al registro anterior.
- Ir al registro siguiente.
- Ir al último registro.

Para agregar un registro nuevo, necesita ir al último registro en la tabla y para ello debe hacer clic en el botón en el último registro y después hacer clic sobre el botón del registro siguiente. Aparecerá una forma en el blanco que puede llenar con sólo introducir la información desde el teclado.

Adición de un registro:

Para agregar un registro, en algunos sistemas debe ir al último registro en la tabla y después hacer clic sobre el botón de registro siguiente o bien, presionar la tecla de flecha hacia abajo. Debido a que cuando esta en el último registro no hay registro siguiente, esta acción indica que quiere agregar un registro nuevo, por ello el DBMS anexa un nuevo registro en blanco y coloca el cursor en el primer campo de la forma. Algunos sistemas tienen una opción de menú para agregar un registro nuevo. Cuando elige esa opción, el sistema le coloca en una forma en blanco.

Uso de un filtro:

Cuando diseña una forma, determina el orden en que se capturan los datos del campo. A este orden se le llama orden de tabuladores, pues en algunos sistemas antiguos el uso de la tecla de tabulador era la única manera de moverse por los campos de una forma de base de datos.

Para comenzar a capturar el registro nuevo basta con que escriba un nombre dentro del campo. Cuando ha terminado de escribir el nombre, puede presionar la tecla de tabulador o la tecla de enter para ir hacia el siguiente campo.

Modificación y eliminación de registros:

Cambiar un registro existente es fácil. Por supuesto, primero debe exhibir el registro que quiere cambiar.

En algunos sistemas, quizá tenga que presionar la tecla de enter para que los cambios tengan efecto; otros le solicitarán que confirme los cambios antes de permitirte ir a otro registro. Para borrar un registro, también debe mostrarlo primero. A continuación selecciona el comando Delete del menú y el registro se borrará. Con el diseño de la forma, especifica si los registros de referencia cruzada, o de no referencia cruzada, en una subforma, se deben borrar junto con el registro en la forma principal. Una vez más, algunos sistemas le solicitarán la confirmación antes de proceder a borrar el registro.

Uso de un filtro.

Al especificar un filtro, permite la revisión exclusiva de aquellos registros que le satisfagan determinados criterios.

Mientras está especificado un filtro, no puede tener acceso a ningún registro que no satisfaga los criterios del filtro. En algunos sistemas, los filtros se guardan como parte de una definición de forma o tabla. En algunos otros, los filtros son completamente independientes de las formas o tablas, y se especifican según se vayan necesitando.

Los filtros proporcionan un método rápido y conveniente para reducir el número de registros que tiene que trabajar, pero no son adecuados para usarse como criterios altamente elaborados. Para conjuntos de criterios más complejos, debe usar consultas.

5.3.2. BUSQUEDA EN BASES DE DATOS

Examinar los registros de una base de datos de uno en uno, no constituyen un método para localizar los datos que necesitas encontrar. Las bases de datos contienen cientos (sino es que miles) de registros, por lo que encontrar un determinado registro de esta manera puede resultar tedioso y tomar mucho tiempo. Afortunadamente, existen herramientas que permiten encontrar un registro específico, rápida y fácilmente.

La mayoría de los programas de base de datos proporcionan dos métodos básicos para encontrar registros: un comando Find o por medio de consultas. El comando Find constituye un método veloz para buscar un registro sin tener que realizar un procedimiento complejo. Las consultas proporcionan mayor versatilidad en los métodos de busca de registros pero son un poco más complejas de diseñar; así que una vez que a resuelto el problema de definir las, las puede guardar para usarlas en el futuro.

Uso del comando Find:

El *comando Find* es empleado comúnmente en la mayoría de los sistemas de base de datos.

SQL:

En el corazón de todo DBMS existe un lenguaje similar a un lenguaje de programación, pero diferente en el sentido de que esta diseñado específicamente para comunicarse con una base de datos. En tiempos pasados,

no había lenguajes estándar. Cada desarrollador de DBMS creó métodos que permitieran a los usuarios la búsqueda y selección de registros en los campos de tablas.

Sin embargo, afortunadamente surgió un lenguaje como estándar. Este lenguaje lo desarrolló originalmente IBM. El lenguaje se denomina SQL. IBM lo desarrolló a finales de los 70's y principios de los 80's como forma para estandarizar los lenguajes de consulta en las muchas plataformas de computadoras de tipo mainframe y minicomputadoras que la compañía producía. SQL se basó en un lenguaje de interrogación previo llamado SEQUEL, que era un acrónimo de "Structured English Query Language"; de ahí la noción común de que SQL quiere decir "Estructurado Query Language (lenguaje de consulta estructurado)", sin embargo, el SQL no quiere decir nada. El nombre derivó solamente de la investigación SEQUEL y hay que apuntar que SQL es cualquier cosa, menos estructurado. De hecho, SQL es la antítesis de estructura. Cuando la organización ANSI lo certificó en 1986, SQL se convirtió en un estándar que ha sido actualizado dos veces con los nombres ANSI SQL-89 y ANSI SQL-92.

El SQL difiere significativamente de los lenguajes de programación. En primer término, la mayoría de los lenguajes de programación todavía son de procedimientos. Un lenguaje de procedimientos consiste de comandos que indican a la computadora lo que debe hacer, instrucción por instrucción, paso a paso. Una declaración SQL en realidad no es un comando para la computadora. En vez de eso, es una descripción de los datos contenidos en

una base de datos. SQL no es de procedimientos porque no da comandos paso a paso a la computadora o a la base de datos; solo describe los datos y en ocasiones pide a la base de datos que haga algo con ellos.

El método más sencillo para explicarle como usar un lenguaje de consulta es decir que simplifica términos y estructuras de lenguaje para interrogar o consultar a la base de datos con respecto a la información que contiene.

En SQL, usted le indica a la base de datos:

- ◆ Con que campos quiere trabajar.
- ◆ La tabla (o tablas) con la cual se trabajarán.
- ◆ Los criterios para seleccionar registros.

Aunque SQL es un estándar ampliamente aceptado no todos los DBMS lo respaldan. Entre las compañías que respaldan al SQL se incluyen Microsoft, Borland, Oracle, Informix e Ingres. Sin embargo, Paradox no respalda directamente a SQL. Para usar SQL en Paradox, el usuario debe comprar un paquete complementario de Borland. No obstante, SQL a permanecido en el mercado desde hace ya tiempo y cada año se abre camino hacia otro producto de base de datos; un ejemplo reciente es Access de Microsoft.

Consulta con ejemplos: INFORMES.

Aunque SQL y los lenguajes de consulta en general son una parte importante de un DBMS, muchos usuarios de PC y Macintosh que trabaja con un DBMS nunca tendrá que escribir una declaración de SQL. Esto se debe a que los desarrolladores de bases de datos proporcionan una interfaz para el usuario para el lenguaje de consulta que recolecta los hechos acerca de una consulta del usuario y componen las declaraciones de SQL o de consulta por sí mismo. Esta característica de interfaz para el usuario se denomina consulta con ejemplo.

Con QBE, usted especifica los criterios de la búsqueda al capturar los valores o símbolos dentro de los campos de una forma de QBE. En algunos sistemas, la formación de QBE luce exactamente igual que una forma en blanco diseñada para capturar datos. En otros, la forma QBE quizás sea una caja de diálogo completamente diferente con una lista de los datos de sus tablas, como es el caso de FoxPro.

Anteriormente, aprendió que una forma es una pantalla que usted diseña para capturar datos y que una forma está enlazada estrechamente con una o más tablas de base de datos. En realidad, una forma también se puede basar en una definición de consulta.

Examinamos la manera en que la gente puede usar un DBMS directamente: es decir, mediante formas para ver y actualizar registros y consultas, para seleccionar registros específicos y crear informes. Sin embargo, mucha gente usa un DBMS sin prestar atención a tablas, campos, formas y

5.3.3. CREACIÓN DE INFORMES.

Así como una forma puede estar basada en una consulta, los informes se basan en consultas. Un informe puede usar una sola tabla o tablas múltiples enlazadas con relaciones de uno a muchos. También pueden incluirse otras tablas en un informe como fuente para otros datos, de manera similar a un campo de búsqueda en una forma.

Diseñar un informe es una labor muy similar al diseño de una forma. Sin embargo, existen algunos criterios adicionales que debe tomar en cuenta. En primer lugar, el orden en que aparecen los registros reviste mayor importancia en un informe que en una forma. El orden puede ser muy importante en informes con cálculos numéricos.

Los campos calculados se pueden incluir en un informe con DBMS que los respalden. En otros sistemas, los cálculos se integran directamente en la definición del informe.

5.4. PROGRAMAS QUE USAN BASES DE DATOS

Examinamos la manera en que la gente puede usar un DBMS directamente; es decir, mediante formas para ver y actualizar registros y consultas; para seleccionar registros específicos y crear informes. Sin embargo, mucha gente usa DBMS sin prestar atención a tablas, campos, formas y

consultas, pues un programa actúa como la interfaz con la base de datos. Ahora veremos algunas de las maneras en que la gente usa la base de datos indirectamente, a través de programas que usan bases de datos.

5.4.1. USO DE BASES DE DATOS DESDE UN PROGRAMA.

Los programas que usan bases de datos pueden ocultar al usuario los detalles de las bases de datos. Los programadores crean un medio en el cual los usuarios pueden trabajar. Dicho medio está diseñado específicamente para permitir a los usuarios concentrarse en las tareas que deben realizar, en vez de concentrarse en el diseño de la base de datos.

Los programas y las bases de datos van junto tan a menudo, que virtualmente cualquier DBMS serio trae incorporado un lenguaje de programación. De hecho, en algunos productos el motor de la base de datos, el lenguaje de consulta y el lenguaje de programación de procedimiento están tan entrelazados que no existen distinciones entre dichos componentes del DBMS. Un ejemplo, es el lenguaje de programación xBASE.

5.4.2. ESTÁNDAR xBASE.

El primer DBMS exitoso y popular para PC fue un programa denominado dBASE. dBASE combinó un lenguaje de programación y de

consulta con un DBMS, y se volvió tan popular que en poco tiempo dBASE se ganó la distinción de ser el DBMS y lenguaje de programación para PC más empleado en el mundo de los negocios. Este éxito condujo a la competencia entre productos de compañías rivales que desarrollaron sistemas similares a dBASE y que usaron el mismo lenguaje de programación y consulta. Uno de estos productos fue Foxbase de Fox Software. Foxbase era completamente compatible con los archivos de dBASE y con el lenguaje dBASE, pero era más veloz. Como resultado, Foxbase obtuvo una porción significativa del mercado para DBMS basados en PC. Dado que Foxbase y otros productos emplearon el lenguaje dBASE, pero no era dBASE, el nombre xBASE se acuñó para describir este lenguaje.

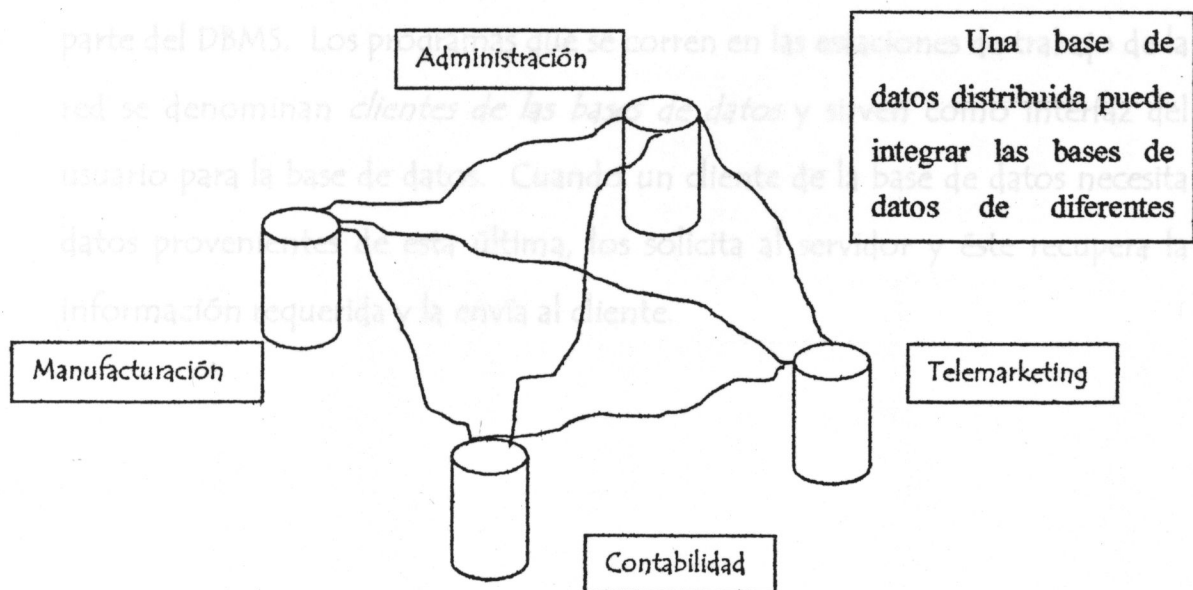
Posteriormente, Borland International compró Ashton Tate, la compañía que producía dBASE. La versión actual es dBASE IV y Borland continúa desarrollando y enriqueciendo al producto. De igual manera, Microsoft compró Fox Software y cambió ligeramente el nombre del producto por FoxPro de Microsoft.

La interfaz original de dBASE se denominó *indicador ("prompt") de punto*. Cuando el usuario corría el programa, éste limpiaba la pantalla, excepto por un carácter puntual o "punto" en la esquina inferior izquierda de la pantalla. Ante el indicador de punto, el usuario tecleaba comandos (incluidos los comandos para crear tablas, formas y reportes, los comandos para usarlas, además de los comandos para crear enlaces entre tablas). Las secuencias de estos comandos se podían guardar en archivos y ejecutar como

un programa. El sistema interpretaría y ejecutaría los comandos. Las versiones posteriores de dBASE incluyeron un sistema de menú para facilitar el empleo a usuarios novatos. FoxPro de Microsoft combina una interfaz de menú y una interfaz de línea de comandos.

5.4.3. BASES DE DATOS DISTRIBUIDAS Y LA RELACIÓN CLIENTE - SERVIDOR.

Un sistema de base de datos distribuido lo integran eficazmente las bases de datos ubicadas en diferentes computadoras. Una base de datos distribuida puede integrar las bases de datos de los diferentes departamentos de una organización o, incluso, las bases de datos en computadoras que estén separadas geográficamente por grandes distancias.



Dado que configurar un sistema de base de datos distribuido puede ser un proceso complejo, esto casi siempre los autorizan los programas.

Una configuración común de red de área local la constituye un grupo de PCs que usan un servidor de archivo común para almacenar programas y datos. En una red, el servidor de archivo es como un ciudadano de relativamente inteligencia escasa. El servidor de archivo espera a que una estación de trabajo de la red le solicite el acceso a un archivo, y cuando eso sucede, el servidor asegura el acceso a ese archivo. Para la estación de trabajo de la red, el servidor de archivo no es más que otra unidad de disco para almacenamiento.

Para las empresas que dependen en gran manera de un DBMS, es esquema más moderno hace un mejor uso del poder de procesamiento de la computadora que almacena los datos. Un *servidor de base de datos* no sólo permite que las computadoras compartan archivos, puesto que ejecuta una parte del DBMS. Los programas que se corren en las estaciones de trabajo de la red se denominan *clientes de las bases de datos* y sirven como interfaz del usuario para la base de datos. Cuando un cliente de la base de datos necesita datos provenientes de esta última, los solicita al servidor y éste recupera la información requerida y la envía al cliente.

5.5 QUE ESPERAR EN EL FUTURO.

A la par que las computadoras de escritorio y los sistemas más grandes se vuelven día con día más poderosos y más fáciles de usar, más gente comenzará a aprovechar las vastas cantidades de información almacenadas en DBMS. Esto se debe a que las computadoras se están volviendo aún más fáciles de usar y a que los DBMS en sí son cada vez más amigables y accesibles, además de que año con año volúmenes crecientes de información están disponibles cada vez más en formatos electrónicos.

Entre la tendencia a observarse en el software para bases de datos se incluye un movimiento continuo hacia ambientes gráficos, tanto para el usuario de software de bases de datos, como para el diseño de bases de datos, formas e informes. Las cada vez más amistosas interfaces de usuario para crear informes y consultas atraerán a usuarios de computadora anteriormente tímidos hacia el mundo de los DBMS. Además son comunes hoy en día los DBMS que permiten que los campos contengan datos binarios, por ejemplo imágenes fotográficas leídas; sin embargo, los campos binarios pueden almacenar algo más que fotografía. Un campo binario puede almacenar vídeo pletórico de sonido y movimiento. Los DBMS se volverán más sofisticados y serán más diversas sus capacidades para usar dichos campos, incluido el "juego" con los datos de vídeo en ellos, ya sea dentro de una ventana o en una forma de base de datos.

Los motores de la base de datos y los lenguajes de consulta también se sofisticarán, al tiempo que los demás estándares evolucionen y sean aceptados por los desarrolladores. Esto podría dar por resultado formatos comunes de archivo de datos, haciendo posibles consultas cruzadas de uno a otro DBMS que puedan conectarse a una red común de bases de datos distribuidas, nacionales o mundiales, para encontrar rápidamente volúmenes de información acerca de casi cualquier tema.

El desarrollo de sistemas es un proceso formado por las etapas de análisis y diseño, comienza cuando la administración o algunos miembros del personal encargado de desarrollar sistemas, detectan un proceso operativo de la empresa que necesita mejorar.

El método del ciclo de vida para desarrollo de sistemas, es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. En la mayor parte de las situaciones dentro de una empresa todas las actividades están muy relacionadas, en general son inseparables, y quizá sea difícil determinar el orden de los pasos que se sigue para efectuarlas. Las diversas partes del proyecto pueden encontrarse al mismo tiempo en distintas fases de desarrollo; algunos componentes en la fase de análisis mientras que otros en etapas avanzadas de diseño.

6

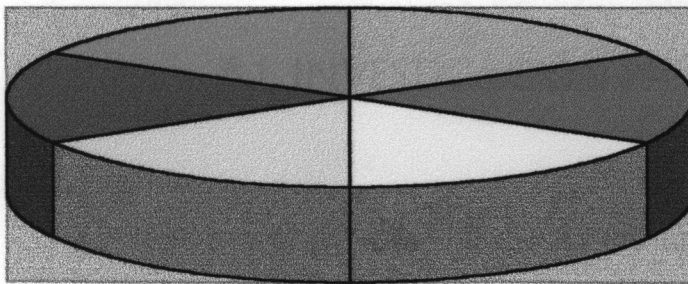
CICLO DE VIDA

El desarrollo de sistemas, es un proceso formado por las etapas de análisis y diseño, comienza cuando la administración o algunos miembros del personal encargado de desarrollar sistemas, detectan un proceso operativo de la empresa que necesita mejorar.

El *método del ciclo de vida para desarrollo de sistemas*, es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. En la mayor parte de las situaciones dentro de una empresa todas las actividades están muy relacionadas, en general son inseparables, y quizá sea difícil determinar el orden de los pasos que se sigue para efectuarlas. Las diversas partes del proyecto pueden encontrarse al mismo tiempo en distintas fases de desarrollo; algunos componentes en la fase de análisis mientras que otras en etapas avanzadas de diseño.

El método del ciclo de vida para desarrollo de sistemas consta de las siguientes actividades:

1. Investigación Preliminar.
2. Determinación de los Requerimientos.
3. Diseño del Sistema.
4. Desarrollo del software.
5. Prueba de los Sistemas.
6. Implantación y Evaluación.



- Investigación Preliminar**
- Determinación de Requerimiento**
- Diseño del Sistema**
- Desarrollo del Sistema**
- Prueba del Sistema**
- Implantación**

Cuando se formula la solicitud comienza el desarrollo de sistemas: la *investigación preliminar*. Esta actividad tiene tres partes: adaptación de la solicitud, estudio de factibilidad y aprobación de la solicitud.

6.1. El método del ciclo de vida para desarrollo de sistemas consta de las siguientes actividades:

1. Investigación Preliminar.
2. Determinación de los Requerimientos.
3. Diseño del Sistema.
4. Desarrollo del software.
5. Prueba de los Sistemas.
6. Implantación y Evaluación.

6.1. INVESTIGACIÓN PRELIMINAR.

La solicitud para recibir ayuda de un sistema de información puede originarse por varias razones; sin importar cuáles sean éstas, el proceso se inicia siempre con la petición de una persona.

Cuando se formula la solicitud comienza la primera actividad de sistemas: la *investigación preliminar*. Esta actividad tiene tres partes: aclaración de la solicitud, estudio de factibilidad y aprobación de la solicitud.

6.1.1. ACLARACIÓN DE LA SOLICITUD.

Muchas solicitudes que provienen de empleados y usuarios no están formuladas de manera clara. Por consiguiente, antes de considerar cualquier investigación de sistemas, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea. Si éste tiene una buena idea de lo que necesita pero no está seguro cómo expresarlo, entonces bastará con hacer una llamada telefónica. Por otro lado, si el solicitante pide ayuda sin saber qué es lo que está mal o dónde se encuentra el problema, la aclaración del mismo se vuelve más difícil. En cualquier caso, antes de seguir adelante, la solicitud de proyecto debe estar claramente planteada.

6.1.2. ESTUDIO DE FACTIBILIDAD.

Un resultado importante de la investigación preliminar es la determinación de que el sistema solicitado sea factible. En la investigación preliminar existen tres aspectos relacionados con el *estudio de factibilidad*:

1. *Factibilidad técnica*: El trabajo para el proyecto puede realizarse con el equipo actual, la tecnología existente de software y el personal disponible? Si se necesita nueva tecnología, ¿cuál es la posibilidad de desarrollarla?
2. *Factibilidad económica*: Al crear el sistema, ¿los beneficios que se obtienen serán suficientes para aceptar los costos?, ¿los costos asociados con la

decisión de no crear el sistema son tan grandes que se debe aceptar el proyecto?

3. *Factibilidad operacional*: Si se desarrolla e implanta, ¿será utilizado el sistema?, ¿existirá cierta resistencia al cambio por parte de los usuarios que dé como resultado una disminución de los posibles beneficios de la aplicación?

6.1.3. APROBACIÓN DE LA SOLICITUD.

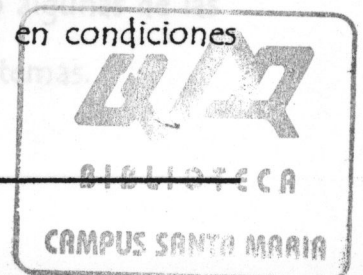
No todos los proyectos solicitados son deseables o factibles. Algunas organizaciones reciben tantas solicitudes de sus empleados que sólo es posible atender unas cuantas. Sin embargo, aquellos proyectos que son deseables y factibles deben incorporarse en los planes. En algunos casos el desarrollo puede comenzar inmediatamente, aunque lo común es que los miembros del equipo de sistemas se encuentren ocupados con otros proyectos. Cuando esto ocurre, la administración decide qué proyectos son los más importantes y decide el orden en que se llevarán a cabo. Muchas organizaciones desarrollan sus planes para sistemas de información con el mismo cuidado con el que planifican nuevos productos y programas de fabricación o la expansión de sus instalaciones. Después de aprobar la solicitud de un proyecto se estima su costo, el tiempo necesario para determinarlo y las necesidades de personal, con esta información se determina dónde ubicarlo dentro de la lista existente de proyectos.

6.2. DETERMINACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA.

El aspecto fundamental del análisis de sistemas es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. Los analistas al trabajar con los empleados y administradores, deben estudiar los procesos de una empresa para dar respuesta a las siguientes preguntas clave:

- ◆ ¿Qué es lo que se hace?
- ◆ ¿Cómo se hace?
- ◆ ¿Con qué frecuencia se presenta?
- ◆ ¿Qué tan grande es el volumen de transacciones o de decisiones?
- ◆ ¿Cuál es el grado de eficiencia con el que se efectúan las tareas?
- ◆ ¿Existe algún problema?
- ◆ Si existe un problema ¿qué tan serio es?
- ◆ Si existe un problema ¿cuál es la causa que lo origina?

Para contestar estas preguntas, el analista conversa con varias personas para reunir detalles relacionados con los procesos de la empresa, sus opiniones sobre por qué ocurren las cosas, las soluciones que proponen y sus ideas para cambiar el proceso. Se emplean cuestionarios para obtener esta información cuando no es posible entrevistar, en forma personal, a los miembros de grupos grandes dentro de la organización. Asimismo, las investigaciones detalladas requieren el estudio de manuales y reportes, la observación en condiciones



reales de las actividades del trabajo y, en algunas ocasiones, muestras de formas y documentos con el fin de comprender el proceso en su totalidad.

Conforme se reúnen los detalles, los analistas estudian los datos sobre requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que deben producir los sistemas junto con características operacionales tales como controles de procesamiento, tiempos de respuesta y métodos de entrada y salida.

6.3. DISEÑO DEL SISTEMA.

El diseño de un sistema de información produce los detalles que establecen la forma en que la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Los especialistas en sistemas se refieren, con frecuencia, a esta etapa como *diseño lógico* en contraste con la de desarrollo del software, a la que denominan *diseño físico*.

Los analistas de sistemas comienzan el proceso de diseño identificando los reportes y demás salidas que debe producir el sistema. Hecho lo anterior se determinan con toda precisión los datos específicos para cada reporte y salida. Es común que los diseñadores hagan un bosquejo del formato o pantalla que esperan que aparezca cuando el sistema esté terminado. Lo anterior se efectúa en papel o en la pantalla de una terminal utilizando para ello algunas de las herramientas automatizadas disponibles para el desarrollo de sistemas.

El diseño de un sistema también indica los datos de entrada, aquellos que serán calculados y los que deben ser almacenados. Así mismo, se escriben con todo detalle los procedimientos de cálculo y los datos individuales. Los diseñadores seleccionan las estructuras de archivo y los dispositivos de almacenamiento, tales como discos y cintas magnéticas o incluso archivos en papel. Los procedimientos que se escriben indican cómo procesar los datos y producir las salidas.

Los documentos que contienen las especificaciones de diseño representan a éste de muchas maneras. La información detallada del diseño se proporciona al equipo de programación para comenzar la fase de desarrollo de software.

Los diseñadores son los responsables de dar a los programadores las especificaciones de software completas y claramente delineadas. Una vez comenzada la fase de programación, los diseñadores contestan preguntas, aclaran dudas y manejan los problemas que enfrentan los programadores cuando utilizan las especificaciones de diseño.

6.4. DESARROLLO DE SOFTWARE.

Los encargados en desarrollar software pueden instalar software comprado a terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, del tiempo disponible para

escribir el software y de la disponibilidad de los programadores. Por regla general, los programadores que trabajan en las grandes organizaciones pertenecen a un grupo permanente de profesionistas. En empresas pequeñas, donde no hay programadores, se pueden contratar servicios externos de programación.

Los programadores también son responsables de la documentación de los programadores y de proporcionar una explicación de cómo y por qué ciertos procedimientos se codifican en determinada forma. La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

6.5. PRUEBA DE SISTEMAS.

Durante la fase de prueba de sistemas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo hagan. Se alimentan como entradas conjuntos de datos de prueba para su procesamiento y después se examinan los resultados. En ocasiones se permiten que varios usuarios utilicen el sistema para que los analistas observen si tratan de emplearlo en formas no previstas. Es preferible descubrir cualquier sorpresa antes de que la organización implante el sistema y dependa de él.

En muchas organizaciones, las pruebas son conducidas por personas ajenas al grupo que escribió los programas originales; con esto se persigue asegurar, por una parte, que las pruebas sean completas e imparciales y, por otra, que el software sea más confiable.

6.6. IMPLANTACIÓN Y EVALUACIÓN.

La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla.

Dependiendo del tamaño de la organización que empleará la aplicación y el riesgo asociado con su uso, puede elegirse comenzar la operación del sistema sólo en un área de la empresa, por ejemplo, en un departamento o con una o dos personas. Algunas veces se deja que los dos sistemas, el viejo y el nuevo, trabajen en forma paralela con la finalidad de comparar los resultados. En otras circunstancias, el viejo sistema deja de utilizarse determinando día para comenzar a emplear el nuevo al día siguiente. Cada estrategia de implantación tiene sus méritos de acuerdo con la situación que se considere dentro de la empresa. Sin importar cuál sea la estrategia utilizada, los encargados de desarrollar el sistema procuran que el uso inicial del sistema se encuentre libre de problemas.

Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses. Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones; realizar cambios y modificaciones en el software, archivos o procedimientos para satisfacer las nuevas necesidades de los usuarios. Dado que los sistemas de las organizaciones junto con el ambiente de las empresas experimentan cambios de manera continua, los sistemas de información deben mantenerse siempre al día. En este sentido, la implantación es un proceso en constante evolución.

La evolución de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

- *Evaluación operacional:*

Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.

- *Impacto organizacional:*

Identificación y medición de los beneficios para la organización en áreas tales como finanzas, eficiencia operacional e impacto competitivo. También se incluye el impacto sobre el flujo de información interno y externo.

- *Opinión de los administradores:*

Evaluación de las actitudes de directivos y administradores dentro de la organización así como de los usuarios finales.

- *Desempeño del desarrollo:*

La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye la valoración de los métodos y herramientas utilizados en el desarrollo.

Desafortunadamente la evaluación de sistemas no siempre recibe la atención que merece. Sin embargo, cuando se conduce en forma adecuada proporciona mucha información que puede ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones subsecuentes.

Se encuentra al final de este documento el disco que contiene el programa con el código fuente, que se realizó para así cumplir con los objetivos y poder también obtener las conclusiones arriba mencionadas.

7

CONCLUSIONES

- ❖ Gracias al sistema se logro un mejor manejo de la información con respecto al control de los clientes.
- ❖ Se tiene con rapidez y sin ningún error de cálculo el presupuesto de la calle que se solicite.
- ❖ Se emiten los reportes sin ningún error en forma rápida, precisa y oportuna.
- ❖ Se agilizó el trámite de búsqueda de la información del cliente que lo solicita.
- ❖ Con el apoyo de esta tecnología, optimizamos el servicio al cliente.
- ❖ Facilita la clasificación de datos en archivo.

- ❖ Se encuentra al final de este documento el disco que contiene el programa con el código fuente, que se realizó para así cumplir con los objetivos y poder también obtener las conclusiones arriba mencionadas.

BIBLIOGRAFIA

Diccionario de Computación.

Freedman.

Mc Graw Hill.

Primera edición.

1973.

Introducción a la Computación.

Norton Peter.

Mc Graw Hill.

Primera edición.

1974.

Análisis y Diseño de Sistemas de Información.

Stall James A.

Mc Graw Hill.

Segunda edición.

1993.

8

BIBLIOGRAFIA

Diccionario de Computación.

Freedman.

Mc Graw Hill.

Primera edición.

1993.

Introducción a la Computación.

Norton Peter.

Mc Graw Hill.

Primera edición.

1994.

Análisis y Diseño de Sistemas de Información.

Senn James A.

Mc Graw Hill.

Segunda edición.

1994.

Elementos de Computación.

Levine Guillermo. García Miguel.

Mc Graw Hill.

Primera edición.

1994.

Visual Fox-Pro 3.0 y 5.0 Manual de Programación.

Pinter Les – Pinter John.

Mc Graw Hill.

Primera edición.

1997.

Estructura de Datos. Introducción a la Base de Datos

Lipschutz Seymour. Ben Begoña.

Mc Graw Hill.

Primera edición.

1987.

Informática Presente y Futuro.

H. Sanders Donald. Lipschutz Seymour

Mc Graw Hill.

Tercera Edición.

1990.

Informática Básica.

Alcalde Eduardo y García Miguel.

Mc Graw Hill.

Segunda Edición.

1994.

Introducción a la Informática.

Fournier.

Megabyte.

Primera Edición.

1992.

Estructura de Datos e Introducción a la Base de Datos.

Albizuri Romero Miren Begoña.

Limusa.

Primera Edición.

1989.

Fundamentos de Base de Datos.

F. Korth Henry. / Silberschatz Abraham

Mc Graw Hill.

Segunda Edición.

1988

Análisis y Diseño Orientado a Objetos.

James Martin / James J. Odell

Prentice – Hall

Primera Edición.

1994.

Análisis de Sistemas de Información.

Home.

<http://fipesmi.misiones.org.ar/users/educohen/analisis.htm>

1998.

ADMINISTRACION DE BASES DE DATOS.

- Persona responsable del diseño físico y de la administración de la base de datos además de la evaluación, selección e implementación del DBMS. En organizaciones pequeñas, el administrador de bases de datos y el administrador de datos son una sola persona; sin embargo, cuando las 2 responsabilidades son administradas en formas separadas, la función del administrador de bases de datos es más técnica.

BASE DE DATOS:

- Colección integrada de datos almacenados en un dispositivo de almacenamiento de acceso directo, serie de información relacionada.

9

APENDICE

BASE DE DATOS DE ARCHIVOS PLANOS:

- Base de datos no relacionales: los datos no están conectados o enlazados a otros datos.

9.1. GLOSARIO DE TERMINOS.

BASE DE DATOS DISTRIBUIDAS:

ADMINISTRACION DE BASES DE DATOS:

- Persona responsable del diseño físico y de la administración de la base de datos además de la evaluación, selección e implementación del DBMS. En organizaciones pequeñas, el administrador de bases de datos y el administrador de datos son una sola persona; sin embargo, cuando las 2 responsabilidades son administradas en formas separadas, la función del administrador de bases de datos es más técnica.

BASE DE DATOS:

- Colección integrada de datos almacenados en un dispositivo de almacenamiento de acceso directo; serie de información relacionada.

- Software de aplicación que le permite al usuario introducir, actualizar y extraer información, así como organizar y buscar esos datos múltiples maneras.
- Conjunto de archivos interrelacionados creado y manejado por un sistema de gestión o de administración de bases de datos (DBMS).
- Cualquier conjunto de datos almacenado en forma electrónica.

BASE DE DATOS DE ARCHIVOS PLANOS:

- Base de datos no relacional; los datos no están conectados o enlazados a otras bases de datos relacionadas.

BASE DE DATOS DISTRIBUIDA:

- Sistema que integra bases de datos localizadas en diferentes computadoras dentro de una red.
- Base de datos que esta físicamente almacenada en dos o más sistemas computacionales. Aunque se encuentra geográficamente dispersa, un sistema de base de datos distribuida administra y controla toda la base de datos como un conjunto único de datos. Si se almacenan datos redundantes en bases de datos separadas, las actualizaciones de un conjunto de datos se hacen de manera automática en los conjuntos adicionales en el momento oportuno.

DBMS (BASE)

- Relacionales por R.C. de Borland. Fue el primer DBMS completo para computadoras personales y aún es el que más se utiliza. Originalmente fue desarrollado por Ashton - Tate, dBASE suministra un criterio

BASE DE DATOS RELACIONAL:

- Base de datos capaz de almacenar tablas; colección de archivos relacionados que comparten por lo menos un campo común; también llamada sistema de administración de bases de datos relacionales.
- Método de organización de base de datos que determina las relaciones entre archivos cuando se requieren. En vez de tener enlaces fijos predeterminados, o punteros entre archivos (clientes a órdenes, proveedores a compras, etc.). una base de datos relacional enlaza los archivos mediante comparación. Este método tiene la flexibilidad de tomar datos o más archivos cualesquiera y generar uno nuevo a partir de los registros que cumplen con los criterios de correspondencia.

BASE DE DATOS / COMUNICACIÓN DE DATOS:

- Se refiere al software que ejecuta funciones de base de datos y de comunicaciones de datos.

BASE DE DATOS 2:

- DBMS relacional de IBM que corre en grandes mainframe. Es un DBMS con todas las características que se ha convertido en el principal producto de base de datos de IBM. Utiliza la interfaz del lenguaje SQL.

DBMS (dBASE):

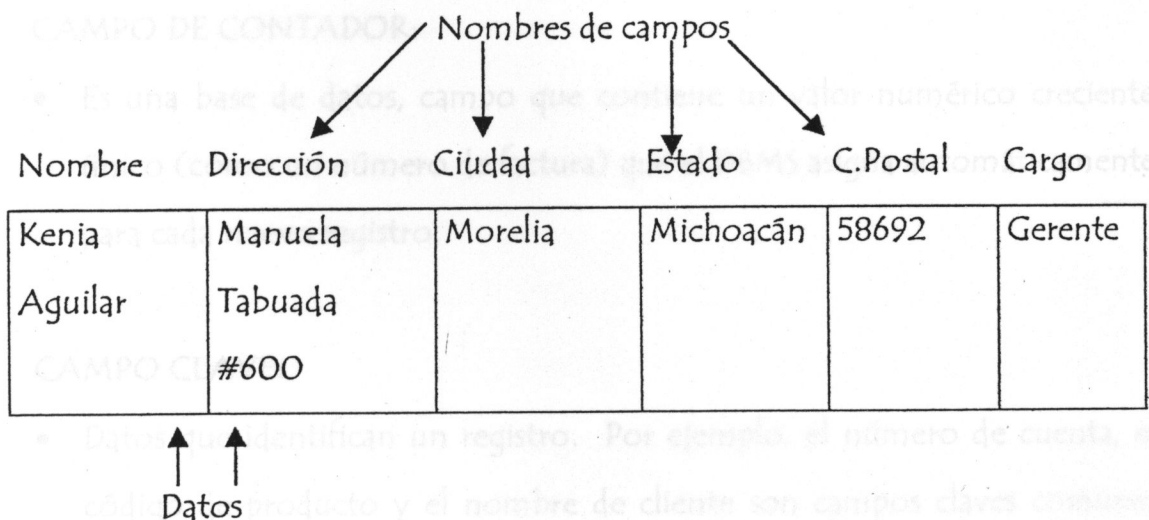
- Relacionales para PC de Borland. Fue el primer DBMS completo para computadores personales y aún es el que más se utiliza. Originalmente fue comercializado por Ashton - Tate, dBASE suministra un entorno

interactivo de base de datos para el usuario y un lenguaje de programación para desarrollar aplicaciones completas. Sus formas de archivos DBF son los estándares de hecho.

- Data Base Management System; sistema de administración de base de datos.

CAMPO:

- Una o más categorías de datos en una base de datos; una columna en una tabla.
- Unidad física de datos que ocupa uno o más bytes. Una colección de campos forman un registro. Un campo también define una unidad de datos en un documento fuente, pantalla o informe. El campo es el común denominador entre el usuario y el computador. Cuando se consulta y actualiza interactivamente una base de datos, se hace referencia a los datos por el nombre del campo. Un campo es la unidad física de almacenamiento, mientras que un dato se refiere al dato en sí mismo.



CAMPO BINARIO:

- Es una base de datos; campo especial usado para guardar un objeto binario grande.
- Campo que contiene números binarios. Puede referirse al almacenamiento de números binarios para propósitos de cálculo o a un campo capaz de contener cualquier información, incluyendo datos, texto, imágenes gráficas, voz y vídeo.

CAMPO DE BÚSQUEDA:

- Es una base de datos, campo que contiene algún factor de identificación que puede ser usado para validar información capturada y simplificar la captura de los datos.

CAMPO CALCULADO:

- Campo numérico o de datos que deriva sus datos del cálculo de otros campos. El usuario no introduce los datos en un campo calculado.

CAMPO DE CONTADOR:

- Es una base de datos, campo que contiene un valor numérico creciente único (como un número de factura) que el DBMS asigna automáticamente para cada nuevo registro.

CAMPO CLAVE:

- Datos que identifican un registro. Por ejemplo, el número de cuenta, el código de producto y el nombre de cliente son campos claves comunes

- utilizados para identificar un registro en un archivo o en una base de datos.
- Como identificador, cada valor de la clave debe ser único para cada registro.

CAMPO ENROLLADO:

- Breve línea en pantalla que puede enrollarse para permitir la edición o exhibición de grandes cantidades de datos en un pequeño espacio de visualización.

CAMPO DE FECHA:

- Campo especial, en una base de datos, que guarda la fecha.

CAMPO DE HORA:

- Campo especial que guarda la hora.

CAMPO LÓGICO:

- Campo que puede contener uno de sólo dos valores, si o no, verdadero o falso, prendido o apagado, menudeo o mayoreo, etc.; también llamado campo sí / no.

CAMPO DE LONGITUD FIJA:

- Por ejemplo, un campo de nombre con 25 bytes ocupa siempre 25 bytes en cada registro.

CAMPO DE LONGITUD VARIABLE:

- Estructura de registro que contiene campos de longitud variable. Por ejemplo, PAT SMITH ocuparía nueve bytes y GEORGINA WILSON quince bytes. También se agregaría un par de bytes de información de control. Si en este ejemplo se usaran campos de longitud fija, para cada nombre deberían reservarse quince o más bytes.

CAMPO MEMO:

- Archivo de datos que contiene una cantidad variable de texto. El texto puede almacenar en un archivo "compañero", pero es tratado como si fuera parte del registro de datos.
- Campo especial que puede contener información de texto de longitud variable, con comentarios o notas.

CAMPO MONETARIO:

- En una base de datos, un campo numérico tiene el formato de despliegue modificado por el software para representar dinero.

CAMPO NÚMÉRICO:

- Campo que contiene números, también llamado campo de números.

CAMPO DE TEXTO:

- Campo que contiene una serie de caracteres alfanuméricos.

REGISTRO:

- En una base de datos, es la fila de una tabla; colección de elementos de datos que pueden ser de longitud fija o variable, uno o más registros generalmente componen un archivo de datos.
- Grupo de campos relacionados que almacenan datos acerca de un tema (registro maestro) o actividad (registro de transacción). Un conjunto de registros constituye un archivo. Los registros maestros contienen datos permanentes, como número de cuenta, y datos variables, como saldo de débito. Los registros de transacciones contienen sólo datos permanentes, como descripción y código del producto.

SISTEMA DE ADMINISTRACIÓN DE BASES DE DATOS (DBMS):

- Software que controla la organización, el almacenamiento, la recuperación, la seguridad y la integridad de los datos en una base de datos. Acepta solicitudes de la aplicación y genera las órdenes al sistema operativo para que transfiera los datos apropiados. Los DBMS pueden ser sistemas autónomos que trabajan con lenguajes tradicionales de programación, como COBOL y C, o pueden ser sistemas completos de desarrollo que incluyen su propio lenguaje de programación y capacidades interactivas para crear y administrar bases de datos, como dBASE y Paradox.

REGISTRO DE LONGITUD FIJA:

- Registro de datos que contiene campos de longitud fija.

REGISTRO MAESTRO:

- Conjunto de datos para un tema individual, como un cliente, empleado o proveedor.

SQL:

- Lenguaje de consulta estructurado.
- Lenguaje utilizado para interrogar y procesar datos en un sistema administrativo de base de datos relacional. Desarrollado originalmente por IBM para sus mainframe, ha habido muchas implementaciones creadas para aplicaciones de bases de datos en mini y microcomputadoras. Los comandos SQL pueden utilizarse para trabajar interactivamente con una base de datos, o pueden incluirse en un lenguaje de programación para servir de interfaz a una base de datos.