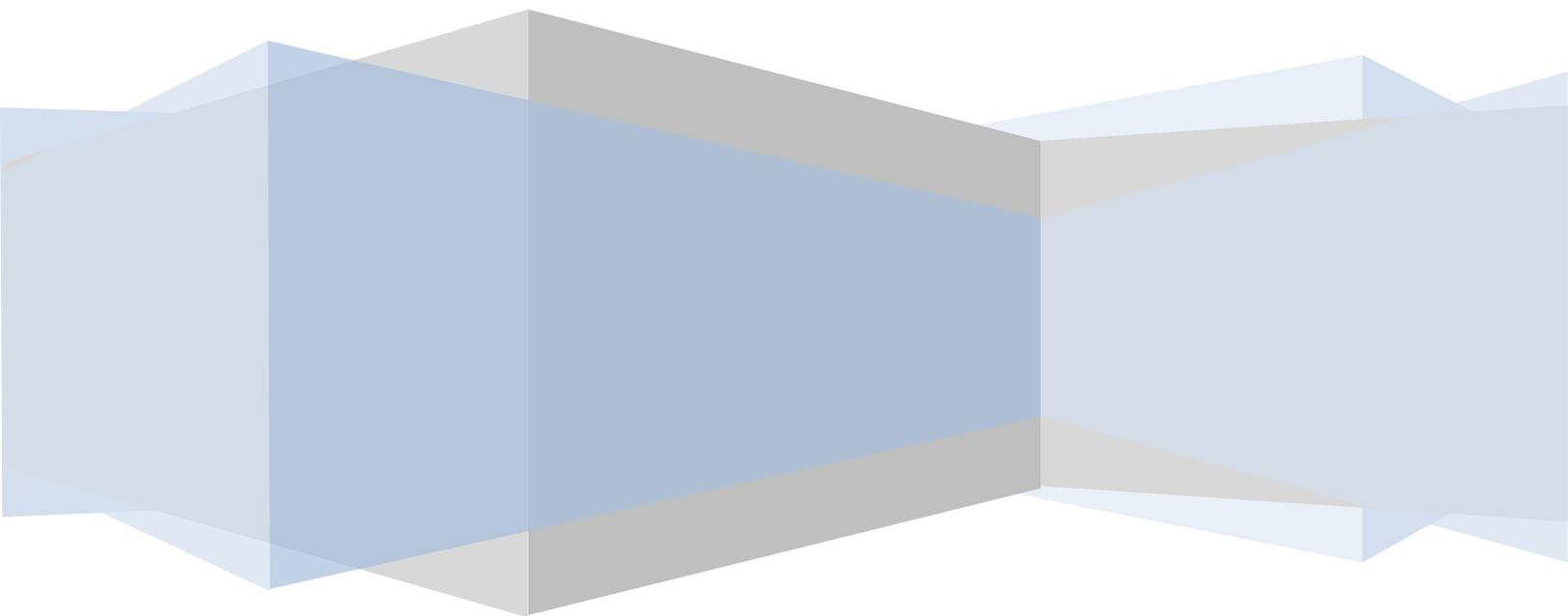


# CAPÍTULO 3

## Bases de Datos





## **CAPÍTULO 3. BASES DE DATOS**

En la actualidad los usuarios cada vez demandan más recursos en cuanto a tecnología, por tanto surgen las evoluciones en los sistemas, y por ende en las bases de datos. Las bases de datos son fundamentales en los sistemas de información debido a que estas se utilizan en todas las áreas profesionales como la investigación, tecnología, arte, educación, sistemas médicos, programas de ingeniería, programas de desarrollo, de diseño, sistemas de información geográfica, entre otros.

### **3.1 Definición de las bases de datos**

Para definir “bases de datos” es importante tener claro qué es un dato e información debido a que estos elementos son fundamentales para el desarrollo de las bases de datos, según (Juárez, 2006, p.45):

- Dato: es un conjunto de caracteres con algún significado, pueden ser numéricos, alfabéticos, o alfanuméricos, este es la unidad mínima de información. Un dato dentro de una base de datos responde a la función (objeto, atributo, valor).
- Información: es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario, para que un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a información, primero se debe guardar lógicamente en archivos.

La información es el recurso más valioso en una base de datos, por tanto esta debe ser:

- Accesible: es la facilidad y rapidez para poder acceder a ella
- Clara: debe ser integra y fácil de entender
- Precisa: lo más exacta posible
- Propia: Debe haber la mayor similitud entre el resultado creado y lo que el usuario pide
- Oportuna: El proceso de entrada-procesamiento-entrega al usuario debe ser en el menor tiempo posible

- Flexible: la información se puede adaptar a la toma de decisiones que mejor convenga
- Verificable: la información debe ser totalmente fiable para que se pueda verificar en el momento deseado
- Imparcial: La información debe poder modificarse tanto por el administrador, como por el usuario dueño de la base
- Cuantificable: la información puede ser el resultado de cualquier dato procesado.

Para llegar a definir “bases de datos”, se debe remontar a lo que eran los archivos y conjuntos de datos, ya que según (Arreguin, 2006, p. 9) se entiende como un archivo a un conjunto de datos interrelacionados, recolectados, que satisfacen las necesidades de información de una comunidad determinada de usuarios, estos son los que dieron auge a la creación de numerosas bases de datos.

Algunos autores han dado varias definiciones de una base de datos, mismas que difieren en el enfoque que se les dé:

Se puede decir que una base de datos es un conjunto de archivos dedicados a guardar información relacionada entre sí, con referencia entre ellos de manera que se complementen y con la posibilidad de relacionarlos en base a diferentes criterios.

La definición anterior es un tanto general, para poder dar una definición más completa es necesario que se consideren algunas de las propiedades de una base de datos, mismas que según (Aguilar, 2006, p. 17) son:

- Representa algún aspecto del mundo real, llamado universo de discurso (UoD, Universe of Discourse) del cual provienen los datos.
- Es un conjunto de datos lógicamente coherente, con significado implícito. Un grupo de datos sin relación entre sí, agrupados de forma aleatoria, no se considera una base de datos.
- Toda base de datos se diseña, se crea y se carga con datos, con un objetivo determinado, y está dirigida a un grupo de usuarios, interesados en el contenido de la base de datos.

Según (Gómez, 2007, p.18) una base de datos es un conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su posterior uso, es una colección de datos estructurados según un modelo que refleje las relaciones y restricciones existentes en el mundo real. Los datos que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción han de ser únicas estando almacenadas junto a los mismos.

Por otro lado (Juárez, 2006, p.45) dice que una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y que ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización.

Lo anterior conduce a que, las bases de datos son un conjunto de información relacionada no redundante (es decir, no debe existir información repetida o duplicada en diferentes tablas), que está organizada, sistematizada y que debe encauzarse en un propósito específico de una comunidad. Del mismo modo tiene que cumplir con los objetivos de independencia (es la capacidad de hacer modificaciones en el modelo físico o lógico sin alterar ninguna aplicación y estructura de datos), integridad (se refiere a la consistencia de los datos y a su vez a que los valores que posean, mismos que deben ser validos de acuerdo a las dependencias funcionales entre las tablas) y la seguridad de datos ante los múltiples usuarios que la utilicen, debido a que cualquier tipo de datos que se utiliza en una base, es de fundamental importancia que no sufra cambios por usuarios que no están debidamente acreditados para hacerlos.

Algunos términos como redundancia, consistencia, independencia, integridad y seguridad se abundarán con detalle más adelante.

### **3.2 Fundamentos históricos y teóricos de las bases de datos.**

Las bases de datos datan desde los años 60, a la fecha, a lo largo de estos años han surgido diversos acontecimientos que permiten mejoras en los diversos software, que han desarrollado diversas compañías dedicadas a la creación de herramientas para el manejo de las bases de datos.

A continuación se presenta una cronología de los eventos más importantes en las bases de datos según (Arreguin, 2006, p. 10) tabla 4 .

Año	Descripción
<b>1960</b>	Uso de los archivos separados ISAM (Index Sequential Access Method) y VSAM (Virtual StorGE Access Method) fueron sistemas administradores de archivos.
<b>1964</b>	El término Base de datos, primero usado en las Fuerzas Militares de EEUU aparece en las publicaciones. IBM introduce IDS (Almacén de Datos Integrado).
<b>1969</b>	IBM introduce IMS, una base de datos jerárquica (con el fin de programas de conquista especial, permite construir cadenas de registros entre ficheros y recorre dichas cadenas). En octubre el DBTG (Database Task Group) envió su primer informe a la PLC, proponiendo un Lenguaje de Descripción de Datos (DLL) para describir una base de datos y un Lenguaje de Manipulación de Datos (DML). Los primeros SGBD (Sistemas Gobernadores de Bases de Datos) se caracterizan por la separación de la descripción de los datos de la manipulación de estos por los programas de aplicación.
<b>1970</b>	E.F. Codd hizo un documento describiendo el modelo relacional (definición de modelo relacional) basado en la simplicidad matemática del álgebra relacional.
<b>1971</b>	Se produjo un informe (DBTG) surgiendo por primera vez dos lenguajes de descripción de datos, uno para el esquema y otro para el subesquema.  Se forma una comisión llamada Comisión de Lenguajes de descripción de Datos (DDLDC).
<b>1972</b>	Finales de 1971 y principios de este año aparece el primer borrador del modelo Codasyl con una arquitectura con una arquitectura de dos niveles: un esquema que proporciona la vista del sistema y un subesquema que proporciona la vista del sistema y un subesquema que proporciona la vista del usuario.
<b>1973</b>	La DDLDC publicó su propuesta en el Journal od Development, reconociendo que le proceso de desarrollo en lenguajes de bases de datos debería ser evolutivo (COBOL).

<b>1976</b>	DER Chen. Surge el SQL. Chen introduce el modelo entidad-vínculo (ER)
<b>1977</b>	Oracle Corp. Se establece. Surge QUEL (Quero Lenguaje) del sistema INGRES. Surge el lenguaje QBE (Quero by Example) desarrollado por IBM.
<b>1978</b>	Oracle hace la primera implementación real del modelo relacional. La comisión técnica X3H2 ha comenzado a estandarizar el modelo de red.  Se lanzó una definición borrador de un esquema de almacenamiento, pero jamás ha sido aprobada ninguna versión final.
<b>1979</b>	Primer RDBMS comercial Oracle V2.
<b>1980</b>	Se funda Informix (3° generación de DBMS), cerca de 10000 usando DBMS en EEUU.  Para PC Paradox. Evolución del modelo relacional.
<b>1981</b>	IBM comenzó a comercializar el SQL.  La pequeña compañía Ashton- Tate introduce DBASE II (no es de DBMS, es de archivos).  Las facilidades de Codasyl incluyen un lenguaje de descripción de datos (DDL) de subesquema y un correspondiente Lenguaje de Manipulación de Datos (DML) Cobol JoD de 1981.
<b>1982</b>	Inician los esfuerzos de ANSI SQL.
<b>1983</b>	Oracle libera sus primera DBMS que corren en mainframes, minicomputadoras y PC's.  IBM lanzó al mercado su producto DATABASE2, más conocido por su abreviatura DB2.  Se disolvió la principal comisión Codasyl que se ocupaba del modelo, porque terminó su objetivo al crear un modelo con éxito, que ANSI tomó a su cargo para estandarizar.

<b>1984</b>	<p>Ashton- Tate introduce DB III.</p> <p>El trabajo sobre el modelo de res (ANSI, 1984 a,b,c y d) basado en el modelo Codasyl, está casi completo, mientras que el modelo relacional (ANSI/SPARC 1984) está todavía en la etapa inicial (Gallagher, 1984).</p>
<b>1985</b>	<p>Comenzaron a conectar las computadoras por medio de redes de área local (LAN).</p> <p>IBM introduce DB2 para maxicomputadoras o mainframes bajo Sistema Operativo MVS.</p>
<b>1986</b>	<p>Informix pasa a ser una compañía pública.</p> <p>Oracle libera su primer DBMS con capacidades de distribución.</p> <p>Se publica el estándar ANSI SQL.</p> <p>El Dr. Date define BD distribuidas.</p> <p>SQL fue adaptado como el estándar ANSI (American National Standarts Institute), para los lenguajes de DB relacionales.</p> <p>Los lenguajes usados son: SQL Structure Quero Language, QUEL Query Language, qbe Query By Example.</p> <p>DB orientadas a objetos.</p> <p>Primera generación Gbase francesa Grápale. Surge la primera generación de SGNDOO (Sistema Gobernador de DB Orientada a Objetos).</p>
<b>1987</b>	<p>ISO (International Standards Organization) adopta el estándar SQL.</p>
<b>1988</b>	<p>Se libera SQL Server para OS/2 por parte de Microsoft y SyBase. Libera Oracle V6.</p> <p>Ashton-Tate introduce DBASE IV. Ontologic lanza Vbase.</p> <p>Symbolics introdujo Static (apoyo a inteligencia artificial como LISP – Lenguaje de procesamiento de Listas).</p>
<b>1989</b>	<p>ANSI-89 SQL. El estándar fue actualizado.</p>

	<p>Se forma SQL Access Group. Se utiliza la arquitectura cliente/servidor y plataforma común.</p> <p>Segunda etapa de SGBDOO.</p>
<b>1990</b>	<p>Ashton-Tate introduce dBase IV.</p> <p>Tercera generación de BD 0.0 (Integridad en forma más rápida y simple), sobre todo en áreas de investigación.</p>
<b>1991</b>	<p>Modelo orientado a objetos (autores: Jeffcoate, Guilfoyle y Deutsch).</p> <p>Oracle alcanza el poder de 1000 TPS en una máquina de cómputo paralelo.</p> <p>Las bases de objetos (también conocidas como bases de datos orientadas a objetos o bases de datos objetuales).</p> <p>El ODMG (Object Management Group, hoy día Object Data Management).</p> <p>El uso creciente de Unified Modeling Language (UML) para la especificación y diseño de sistemas de información propicia que sus datos se gestionen en bases de objetos de una manera más natural (tecnología de objetos del principio al final) que es SGBDs.</p>
<b>1992</b>	<p>ANSI – 92 SQL</p> <p>Triggers (disparador: acciones ejecutadas automáticamente).</p> <p>SQL Access Group publica las especificaciones CLIENTE.</p> <p>Microsoft libera ODBC 1.0 para Windows.</p>
<b>1993</b>	<p>Oracle implanta DB distribuidas.</p> <p>Access puede trabajar con datos dependientes de dBase, Paradox, Foxpro.</p> <p>Paradox 4.0 de Borland anuncia PAL (Paradox Applications Language for Windows).</p> <p>SyBase anuncio la versión 10.0, una versión completamente nueva de SQL Server y sin</p>

	<p>intervención de Microsoft.</p> <p>Libera Oracle 7 para Unix. Se desarrolla un Ambiente de Desarrollo Cooperativo (CDE) de Oracle.</p> <p>Microsoft libera su versión de SQL Server de 32 para Windows NT.</p>
<b>1994</b>	<p>FoxPro Microsoft.</p> <p>Microsoft libera ODBC 2.0 para Windows.</p> <p>Liberan Oracle 7 para PC.</p> <p>En abril Microsoft anunció que se desarrollaría su propio producto servidor de bases de datos. Las versiones posteriores de Microsoft SQL Server traían recuerdos de SQL Server de SyBase. Microsoft permite portar las especificaciones de ODBC a otras plataformas no Windows por parte de Visigenic Software.</p> <p>ODBC 2.0 SDK es disponible para los UNIX de SUN, HP e IBM.</p> <p>SQL Access Group se fusiona con X/Open.</p>
<b>1995</b>	<p>Se anuncia DB2/2 de IBM.</p> <p>Oracle 7.1.</p> <p>SQL Server 6.0 de Microsoft (puede trabajar con Access, Lotus Approach y Borland Paradox).</p> <p>En este año Informix tenía el 34.5% del mercado RDBMS para UNIX.</p> <p>El SDK de ODBC está disponible para SCO UNIX, OS/2, Macintosh y Power Macintosh.</p>
<b>1996</b>	<p>Mejor DB Oracle 7.2.</p>

	<p>Oracle Parallel Server incrementa performance introduciendo operaciones en paralelo y asíncronas.</p> <p>Oracle Universal Server transfiere desde PC a poderosos servidores de redes.</p> <p>Oracle 8 está en versión Beta.</p> <p>Javasoft está preparando una especificación para unir aplicaciones Java con Bases de Datos relacionales.</p> <p>La subsidiaria de Sun Microsystems dispone de una herramienta que permite unir aplicaciones Java a BD ODBC. Java Database Connectivity (JDBC) crea un cambio uniforme para que las aplicaciones Java accedan a diferentes bases de datos SQL.</p>
<b>1997</b>	<p>Herramientas OLAP (On Line Analytical Processing) ventajas de BD en bloques de tres dimensiones. Oracle se encuentra en la versión 8.i, SQL server de SyBase en la 11 y SQL Server de Microsoft en la 6.5.</p> <p>Se introducen nuevos conceptos en las bases de datos como: Datawarehousing (formulada por bases de datos que pueden manejar una cantidad enorme de información), bases de datos orientadas a objetos, JDBC (Java Data Base Connectivity) éste en un API (Interface de Aplicación) que une a Java con la Bases de datos.</p>
<b>1999</b>	<p>Se publica SQL2000 en el que se agregan expresiones regulares, consultas recursivas (para relaciones jerárquicas), triggers y algunas características orientadas a objetos.</p>
<b>2001</b>	<p>Command Prompt, Inc. lanzó Mammonth PostgreSQL, la más antigua distribución comercial de PostgreSQL.</p>
<b>2003</b>	<p>SQL introduce algunas características de XML, cambios en las funciones, estandarización del objeto sequence y de las columnas autonuméricas.</p>
<b>2006</b>	<p>La norma ISO/IEC 9075-14:2006 define las maneras en las cuales el SQL se puede utilizar conjuntamente con XML. Define maneras de importar y guardar datos XML en</p>

	una base de datos SQL. Facilita el manejo de aplicaciones para integrar es uso de XQuery dentro del código SQL, para el uso concurrente a datos ordinarios.
2007	EnterpriseDB anunció el Postgres Resource Center y EnterpriseDB Postgres, diseñados para ser una completamente configurada distribución de PostgreSQL incluyendo muchos módulos contribuidos y agregados
2008	Se introduce el uso de la cláusula ORDER BY en SQL2008

*Tabla 4. Cronología de las bases de datos*

### *Conceptos básicos*

Para una comprensión más clara de las bases de datos es necesario identificar los tecnicismos que se utilizan de manera cotidiana en el ambiente de las bases de datos, dentro de los conceptos básicos están:

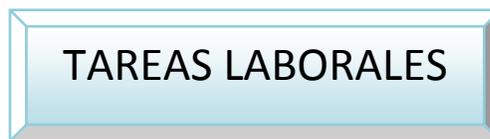
- **Entidad:** persona, lugar, objeto o evento de interés acerca del cual se recogen o procesan datos. Esta se representa por medio de un rectángulo, mismo que contiene dentro el nombre de la entidad.



*Figura 10. Identidad.*

Las entidades se dividen en dos tipos, que son:

- Regulares o fuertes, que son las entidades normales que tienen existencia por sí mismas sin depender de otras (su representación gráfica es la mostrada en la figura anterior).
- Débiles, su existencia depende de otras. Su representación gráfica es con doble recuadro.



*Figura 11. Identidades.*

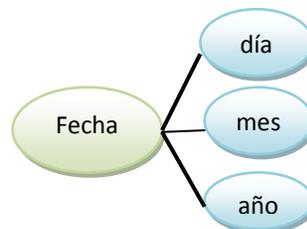
- **Atributo:** es una característica de una entidad o de una relación en función de lo que nos interesa en nuestra aplicación. Su representación gráfica es una elipse.



*Figura 12. Atributo.*

Los atributos se clasifican en:

- Simples, no están divididos en subpartes
- Compuestos, estos están divididos en subpartes, una forma clara de ejemplificar este tipo de atributos es, por ejemplo el atributo fecha, este es un atributo compuesto, ya que fecha se subdivide en tres atributos como día, mes y año.



*Figura 13. Atributos.*

- Monovaluados, solo pueden tener un valor para una entidad particular.
- Multivaluados, pueden tener más de un valor para una entidad.



*Figura 14. Atributos multivaluados.*

- Almacenados o derivados, son atributos cuyo valor para una entidad puede obtenerse en función de los valores almacenados en otros atributos. Su representación gráfica es mediante una elipse con línea discontinua.

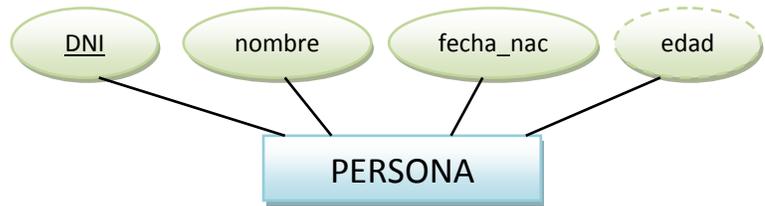


Figura 15. Entidad.

- **Identificador**, Se trata de uno o más campos cuyos valores son únicos en cada ejemplar de una entidad. Se indican subrayando el nombre del identificador.

Para que un atributo sea considerado buen identificador tiene que cumplir:

- ✓ Deben distinguir a cada ejemplar teniendo en cuenta las entidades que utiliza el modelo. No tiene que ser un identificador absoluto
- ✓ Todos los ejemplares de una entidad deben tener el mismo identificador
- ✓ Cuando un atributo es importante aun cuando no tenga una entidad concreta asociada, entonces se trata de una entidad y no de un atributo.

Es importante señalar que al conjunto de valores que puede tomar un atributo se le llama *dominio* del atributo.

- Toda entidad debe tener al menos un atributo que permita diferenciar unas entidades particulares de otras, es decir que no toman nunca el mismo valor para dos entidades particulares diferentes. A estos atributos se les llaman **claves**, la representación gráfica de las claves se señala subrayando la palabra que etiqueta al atributo.

Las claves también ayudan a identificar unívocamente a las relaciones y así a distinguir relaciones entre sí.

- Superclave, es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades.
  - Clave candidata, son también superclaves mínimas debido a que son superclaves tales que sus subconjuntos no son superclaves.
  - Clave primaria, es la clave elegida por el diseñador de la base de datos como elemento principal para identificar las entidades dentro de un conjunto de entidades. Esta clave se debe elegir de manera que sus atributos nunca, o muy raramente cambien.
- **Tabla**, es un conjunto de datos dispuesto en una estructura de filas y columnas. En una tabla las filas se denominan registros y las columnas campos; la primera fila contiene los nombres de campo. Cada campo contiene determinado tipo de datos y tiene una longitud expresada en el número de caracteres máximo del campo. Para crear una tabla es necesario definir su estructura:
    - El nombre de la tabla
    - Los tipos de dato de cada campo
    - Las propiedades o características de cada campo
    - El campo clave

Los datos pueden ser de diversos tipos.

Tipo de dato	Almacena	Tamaño
<b>Texto</b>	Caracteres alfanuméricos	Hasta 255 bytes
<b>Memo</b>	Caracteres alfanuméricos	Hasta 64000 bytes
<b>Numérico</b>	Valores enteros o fraccionarios	1,2,4 u 8 bytes
<b>Fecha/Hora</b>	Fechas y horas	8 bytes
<b>Moneda</b>	Valores de moneda	8 bytes
<b>Autonumerado</b>	Valor numérico de incremento automático	8 bytes
<b>Si/No</b>	Valores Booleanos (verdadero / falso)	1 byte
<b>Objeto OLE</b>	Imágenes o gráficos	Hasta 1Gb.

*Tabla 5. Tipos de datos en una base de datos.*

Es importante señalar que dentro de los datos de tipo numérico se clasifican en:

Valor de campo	Rango	Lugares decimales	Tamaño
<b>Byte</b>	0-255	Ninguno	1 byte
<b>Entero</b>	-32.768 a 32.767	Ninguno	2 bytes
<b>Entero largo</b>	-2.147.486.648 a -2.147.486.647	Ninguno	4 bytes
<b>Simple</b>	$-3.4 \cdot 10$ a $3.4 \cdot 10$	7	4 bytes
<b>Doble</b>	$-1.797 \cdot 10$ a $1.797 \cdot 10$	15	8 bytes

*Tabla 6. Clasificación de los datos tipo numérico.*

Según (Juárez, 2006, p. 47) en una tabla es de fundamental importancia identificar las propiedades de un campo como se mostrarán en la siguiente tabla, mismas que son de gran ayuda para el manejo de la apariencia que tienen los campos, evitar la introducción incorrecta de los mismos, especificar valores predeterminados, acelerar la búsqueda y ordenar una tabla mediante índices.

Propiedades del campo	Utilidad
<b>Tamaño de campo</b>	Ajusta el tamaño de un campo de tipo texto o limita el rango de valores permitidos en un campo numérico
<b>Máscara de entrada</b>	Presentación de un formato de un campo para no escribirlos y los datos se ajusten a la máscara
<b>Reglas de validación</b>	Permite limitar los datos introducidos en un campo
<b>Texto de validación</b>	Texto que se muestra al infringir la regla de validación

*Tabla 7. Propiedades de un campo.*

- **Relaciones:** son el elemento del modelo de bases de datos que permite relacionar en sí los datos del mismo. Son también vistas como una tabla, ya que tiene como atributos los nombres de las columnas de una tabla, y también cada tupla corresponde a una fila de la tabla, para el caso de las bases de datos según (Martínez, 2006, p. 44) una tupla es la que representa un conjunto de valores relacionados permanentemente, dicho conjunto puede representar tanto entidades como relaciones. Por lo tanto se puede decir que una relación describe la interacción entre dos o más entidades. Su representación gráfica es mediante un rombo.



*Figura 16. Relación.*

Un ejemplo gráfico de una relación es:



*Figura 17. Relación.*

Dependiendo del número de entidades que se relacionan, o de la relación que se denota por medio de líneas que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones, dependiendo de lo anterior existirá una relación binaria, doble, ternaria o reflexiva.

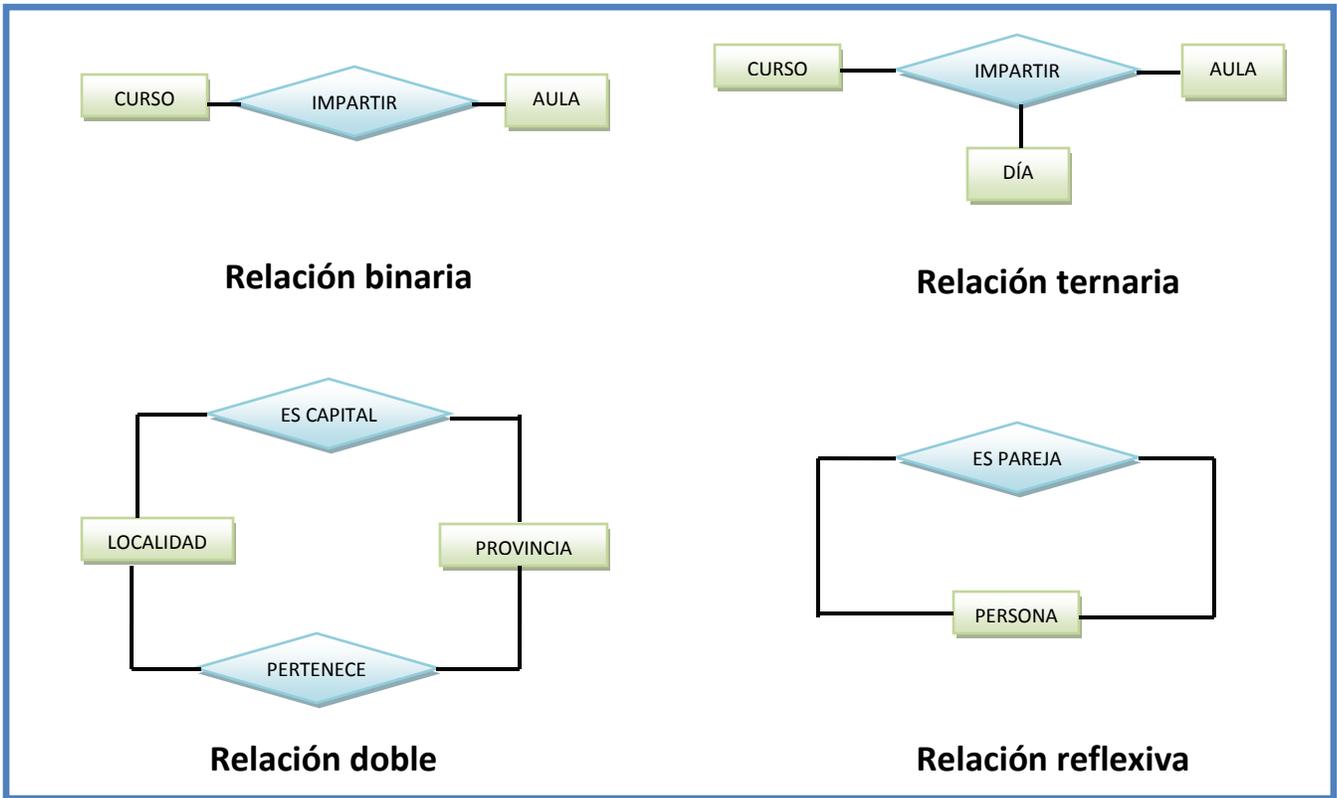


Figura 18. Relaciones de bases de datos.

- **Conjunto de entidades**, que son el conjunto de entidades que comparten las mismas propiedades o atributos.

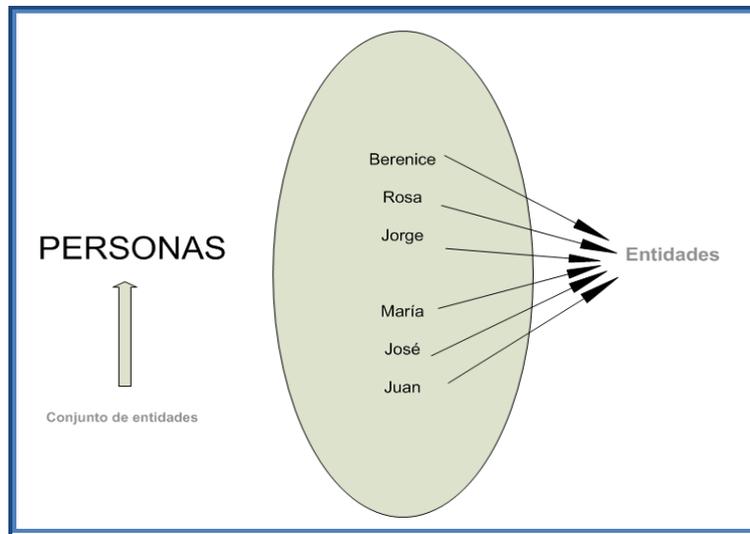
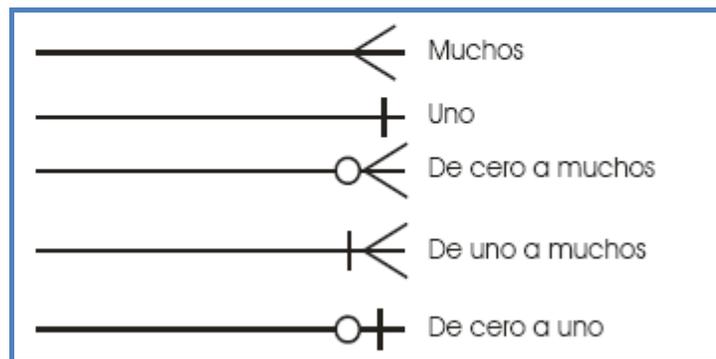


Figura 19. Relaciones de entidades.

- **Conjunto de relaciones**, mismo que se forma con relaciones del mismo tipo.
- **Cardinalidad**, indica el número de relaciones en las que una entidad puede aparecer, se anota en términos de :

- Cardinalidad mínima, indica el número de asociaciones en las que aparecerá cada ejemplar de la entidad (el valor que se anota es de cero uno)
- Cardinalidad máxima, indica el número máximo de relaciones en las que se puede aparecer cada ejemplar de la entidad (puede ser uno o muchos)

La cardinalidad se indica de varias formas tales como:



*Figura 20. Cardinalidad.*

Las herramientas conceptuales para la descripción de datos, relaciones entre datos, semántica de los datos y restricciones de consistencia, son la base para comprender la estructura de una base de datos. La simplicidad conceptual ha permitido que usuarios no especializados en esta área puedan entender un simple esquema de base de datos.

### *Características de las bases de datos*

Según (Navarro, 2009, p. 15) una base de datos brinda una buena administración de datos si cumple con las siguientes características:

- *Redundancia*

La redundancia se refiere a la existencia de información repetida o duplicada en diferentes tablas dentro de una base de datos.

Para evitar la redundancia es necesario identificar los datos que son utilizados en común por varias aplicaciones y almacenarlos una sola vez, aunque debemos ser cuidadosos ya que existen excepciones en las que no podemos eliminar del todo la redundancia por que puede existir una sola copia de atributos que son necesarios para diversos fines, porque es necesario almacenar y mantener las relaciones pertinentes.

- *Consistencia*

Frecuentemente los problemas de consistencia de datos se deben a la redundancia de éstos.

Es muy probable que surjan incongruencias al almacenar la misma información en más de un lugar, ya que al modificar, eliminar o agregar un dato, en esas condiciones, debe realizarse en cada una de las instancias del mismo con el riesgo de no realizarlo en su totalidad, generando en este caso datos inconsistentes.

- *Integridad*

La integridad en una base de datos se refiere, a que los valores que posean los datos sean validos de acuerdo a las dependencias funcionales de las tablas. Esto es que cuando una base de datos incluya información utilizada por muchos usuarios, es importante que no puedan destruirse los datos almacenados ni las relaciones que existen entre los distintos atributos.

La integridad de la base de datos se puede lograr mediante:

- El mantenimiento de una redundancia mínima y controlada,
- el establecimiento de llaves primarias o índices primarios,
- la validación de las dependencias entre las tablas relacionadas, y

- la creación de reglas de validación durante la inserción y edición de datos.
- *Seguridad*

La seguridad implica asegurar que los usuarios están autorizados para llevar a cabo lo que tratan de hacer.

La seguridad de una base de datos se refiere principalmente al control de acceso, modificación y definición, tanto de los datos como de la estructura de la base de datos por parte de los diferentes usuarios de la misma.

Según (Maldonado, 2007, p. 7) la seguridad implica asegurar que los usuarios estén autorizados para llevar a cabo lo que tratan de hacer.

Entre las formas de acceso malintencionado se encuentran:

- La lectura no autorizada de los datos (robo de información)
- La modificación no autorizada de los datos
- La destrucción no autorizada de los datos

Para proteger la base de datos se deben adoptar medidas de seguridad en varios niveles:

- Sistema de base de datos, es responsabilidad del sistema de bases de datos asegurarse de que no se olviden las restricciones de autorización
- Sistema operativo, la debilidad de la seguridad del sistema operativo puede servir como medio para el acceso no autorizado a la base de datos
- Red, dado que casi todos los sistemas de base de datos permiten el acceso remoto, la seguridad en el nivel del software de la red es tan importante como la seguridad física, tanto en internet como en las redes privadas.
- Físico, los sitios que contienen los sistemas informáticos deben estar protegidos físicamente contra la entrada de intrusos
- Humano, los usuarios deben ser autorizados cuidadosamente para reducir la posibilidad de que alguno de ellos dé acceso a intrusos a cambio de sobornos u otros favores.

### ***Modelos de datos***

La abstracción de datos es un nivel característico en una base de datos, ya que es fundamental que los usuarios solo conozcan los datos que le son útiles y no todas las características sobre el almacenamiento físico.

Es importante identificar que un modelo es una representación de la realidad que contiene las características generales de algo que se va a realizar.

Según (Navarro, 2009, p.16) un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Según (Estrada, 2005, p.47) un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas y actualizaciones de datos.

Los modelos de datos se pueden clasificar dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos.

Según (Juárez, 2006, p.56) existen tres modelos dentro de una base de datos estos son:

- **Modelo conceptual**, en este se construye un esquema conceptual de la información que se usará en la base de datos, al construir este esquema, se descubre el significado de los datos, se encuentran entidades, atributos y relaciones. En este modelo se tiene como objetivo comprender:
  - La perspectiva que el usuario tiene de los datos
  - La naturaleza de los datos, independientemente de su representación física

El esquema conceptual se construye utilizando la información que se encuentra en la especificación de los requisitos del usuario. El modelo conceptual es completamente independiente de los aspectos de implementación, como pueden ser los programas de

aplicación, los lenguajes de programación, etc. En este modelo siempre se valida que los requisitos del usuario se cumplan de manera satisfactoria.

- **Modelo lógico**, este modelo es una fuente de información para el diseño físico, transformando al esquema obtenido en el modelo conceptual. El avance de este modelo depende de la evolución que se obtiene al desarrollar, probar y validar los requisitos del usuario. Este modelo es de fundamental importancia en la etapa del mantenimiento del sistema, ya que permite que los futuros cambios que se realicen en los programas de aplicación o sobre los datos, se representen correctamente en la base de datos.
- **Modelo físico**, en éste se produce la descripción de la implementación de la base de datos en memoria secundaria: estructuras de almacenamiento y métodos de acceso que garanticen un acceso eficiente a los datos. Es fundamental que se tenga un buen diseño del modelo lógico, debido a que entre el diseño físico y el lógico existe una realimentación, ya que algunas de las decisiones que se toman durante el diseño físico, pueden afectar a la estructura del modelo lógico.

El objetivo del modelo físico es describir cómo se va a implementar físicamente el esquema formado en el modelo conceptual y lógico, dicha implementación consiste en:

- Obtener un conjunto de relaciones y las restricciones que se deben cumplir sobre ellas
- Determinar las estructuras de almacenamiento y los métodos de acceso que se van a utilizar para conseguir unas prestaciones óptimas
- Diseñar el modelo de seguridad del sistema.

Los distintos modelos se dirigen a usuarios diferentes, por ejemplo los conceptos de los modelos físicos están dirigidos al personal informático y no a los usuarios finales como es el caso de los modelos lógicos.

### ***Modelos conceptuales***

Según (Navarro, 2009, p. 16) se usan para describir datos, es decir con este modelos se representan los datos tal y como se captan del mundo real, tiene la capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente.

Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

- **Modelo Entidad-Relación**

Según (Sánchez, 2004, p. 17) fue creado por Peter Chen en los años 1976 y 1977 a través de dos artículos. Se trata de un modelo que sirve para crear esquemas conceptuales de bases de datos, es un estándar para crear dichos esquemas.

Según (Silberschatz, 2002, p. 5) el modelo de datos Entidad –Relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos como entidades y relaciones, mediante un diagrama E-R se puede expresar gráficamente la estructura lógica general de una base de datos. El diagrama entidad relación consta de diversos componentes como entidades, atributos, relaciones entre conjuntos de entidades, unión de atributos con los conjuntos de entidades y/o relaciones, etcétera.

Cada componente siempre se señala con la etiqueta o relación que representa. En un modelo E-R además de entidades y relaciones, representa ciertas restricciones que los contenidos de la base de datos deben cumplir. Una restricción importante es la correspondencia de cardinalidades, misma que expresa el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones.

- **Modelo relacional**, es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de “relaciones”, las cuales podrían considerarse en forma lógica como tuplas mismas que según (Estrada, 2005, p. 71) son un conjunto de datos.

Según (Navarro, 2009, p. 18) en este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

En este modelo se hace uso del concepto de clave primaria, mismo que nos ayudará a definir un campo de la tabla como atributo principal.

Haciendo una tabla que contenga cada una de las llaves primarias involucradas en una relación, por ejemplo, tomando en cuenta que RFC es la clave primaria y a su vez la llave primaria del artículo es la clave.

La relación resultaría de la siguiente manera representada en una tabla transitiva:

RFC	Clave
PECJ500922XYZ	C001
MEAN761014ABC	B300

*Figura 21. Representación de una relación en una tabla transitiva.*

Según (Gómez, 2007, 71) en este modelo, el lugar y la forma en que se almacenen los datos no tiene relevancia. Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico a la base de datos. La información puede ser recuperada o almacenada mediante “consultas” que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL (Structured Query Lenguaje), un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

En el modelo relacional la base de datos según (Gómez, 2007, p. 55) es percibida por el usuario como un conjunto de tablas. Esta percepción es solo a nivel lógico, ya que a nivel físico puede estar implementada mediante distintas estructuras de almacenamiento.

Cada relación es pensada como si fuese una tabla que está compuesta por un registro (las filas de una tabla) que representan las tuplas, y los campos (las columnas de una tabla).

- **Modelo jerárquico**, en este modelo las bases de datos se almacenan en una estructura jerárquica, en donde un “nodo padre” de información puede tener “varios hijos”. El nodo que no tiene padres es llamado “raíz”, y los nodos que no tienen hijos se los conoce como “hojas”.

Según (Gómez, 2007, p. 72) las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

- **Modelo de red**, este modelo representa los datos mediante colecciones de registros, sus relaciones se representan por medio de ligas o enlaces. Es importante identificar que un registro es similar a una entidad. Según (Navarro, 2009, p. 17) un registro es una colección de campos (atributos), cada uno de los cuales contiene solamente almacenado un valor, el enlace en la asociación entre dos registros exclusivamente, se puede ver como una relación estrictamente binaria.

Según (Gómez, 2007, p. 73) este modelo es ligeramente distinto al modelo jerárquico, con la diferencia fundamental en la modificación del nodo, ya que se permite que un mismo nodo tenga varios padres. También ofrece una solución eficiente al problema de redundancia de datos, pero aun así, la dificultad de administrar la información en una base de datos de red solo es manejada por los programadores y no por los usuarios finales.

Por otro lado, (Estrada, 2005, p. 55) menciona que los datos son representados como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos, que son punteros en la implementación física. Los registros se organizan como un grafo.

- **Modelo orientado a objetos**, define a una base de datos en términos de objetos, sus propiedades y sus operaciones. Este se basa en el concepto de encapsulamiento de datos, el cual se abundará más adelante y código que opera sobre estos en un objeto. Los objetos estructurados se agrupan en clases, es decir los objetos con la misma

estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos denominados métodos.

Para comprender el funcionamiento de este modelo es necesario conocer tres conceptos importantes del paradigma de objetos:

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

Según (Gómez, 2007, p.74) en bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos. La implementación de operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

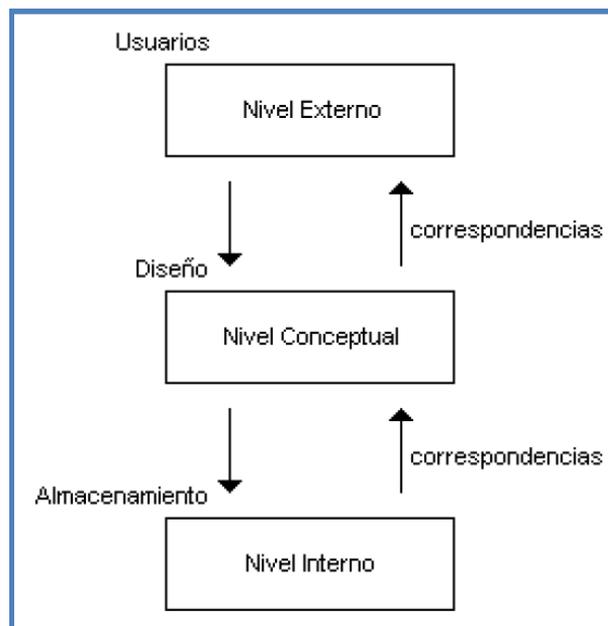
### ***Arquitectura de una base de datos***

Existen tres características importantes en las bases de datos, entre ellas la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de catálogos para almacenar los esquemas de datos. En 1975, el comité ANSI-SPARC (American National Standard Institute – Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles, mismo que resulta útil utilizar para mantener la independencia en la base de datos.

Según (Gómez, 2007, p. 68) una base de datos se utiliza para cualquier tipo de datos, y está construida por tres niveles principales, que no varían por la visión del usuario, ni por el uso que este le dé.

- Nivel interno: es el nivel más bajo de abstracción, y define cómo se almacenan los datos en el soporte físico, así como los métodos de acceso.
- Nivel conceptual: es el nivel medio de abstracción. Se trata de la representación de los datos realizada por la organización, que recoge las vistas parciales de los requerimientos de los diferentes usuarios y las aplicaciones posibles. Se configura como visión organizativa total, e incluye la definición de datos y las relaciones entre ellos.
- Nivel externo: es el nivel de mayor abstracción. A este nivel corresponden las diferentes vistas parciales que tienen de la base de datos los diferentes usuarios. En cierto modo, es la parte del modelo conceptual a la que tienen acceso.

A continuación se muestra el diagrama que según (Gómez, 2007, p. 69) explica por sí mismo la relación que existe entre los niveles de la arquitectura de una base de datos.



*Figura 22. Arquitectura de una base de datos.*

Es de fundamental importancia señalar que el modelo de arquitectura permite establecer el principio de independencia de los datos, esta independencia puede ser lógica y física, por independencia lógica se entiende que los cambios en el esquema lógico no deben afectar a los esquemas externos que no utilicen datos modificados. Por *independencia física* se entiende que el esquema lógico no se va a ver afectado por cambios realizados en el esquema interno, correspondientes a modos de acceso, etc.

La independencia lógica es útil si se requiere ampliar o reducir la base de datos, y la independencia física se utiliza sólo en la separación entre las aplicaciones y las estructuras físicas de almacenamiento.

Por otro lado, (Estrada, 2005, p. 49) menciona lo siguiente:

- En el nivel interno se describe la estructura física de la base de datos mediante un esquema interno. Este se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.

En el nivel conceptual se describe la estructura de toda la base de datos para una comunidad de usuarios, mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir las propiedades de la base de datos.

En el nivel externo se describen varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo determinado de usuarios y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

### ***Tipos de bases de datos***

Las bases de datos se clasifican de acuerdo a distintos criterios como son:

- Variabilidad de datos almacenados
- Contenido

Según (Gómez, 2007, p. 70) existen dos tipos de bases de datos dentro de la clasificación por variabilidad de los datos almacenados.

- Bases de datos estáticas: son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, para así realizar proyecciones y tomar decisiones.
- Bases de datos dinámicas: son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta.

Por otro lado (Navarro, 2009, p. 7) señala que las bases de datos clasificadas de acuerdo a su contenido.

- Bases de datos bibliográficas: Solo contienen un representante de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información de una determinada publicación sobre el autor, fecha de publicación, editorial, título, edición, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino se estaría en presencia de una base de datos a texto completo (o de fuentes primarias).
- Bases de datos numéricas: son bases de datos que su contenido se limita a cifras o números.
- Bases de datos de texto completo: almacenan fuentes primarias.
- Directorios
- Banco de imágenes, audio, video, multimedia, etc.
- Bases de datos o “bibliotecas” de información biológica: almacenan diferentes tipos de información provenientes de las ciencias de la vida o médicas, estas se subclasifican en
  - ✓ Las que almacenan secuencias de nucleótidos o proteínas
  - ✓ Las bases de datos de rutas metabólicas
  - ✓ Bases de datos de estructura, mismas que comprenden los registros de datos experimentales sobre estructuras 3D.
  - ✓ Bases de datos clínicas
  - ✓ Bases de datos bibliográficas (biológicas)

### 3.3 Bases de datos relacionales.

El modelo de datos relacional es importante porque ofrece distintos procesos de datos como; simplicidad y generalidad, facilidad de uso para el usuario final, tiempos cortos de aprendizaje y las consultas de información se especifican de forma sencilla.

El primer paso para crear una base de datos relacional, según (Martínez, 2006, p. 47) es planificar el tipo de información que se quiere almacenar, teniendo en cuenta dos aspectos:

- La información disponible
- La información que necesitamos

Para la administración de la base de datos es de fundamental importancia la planificación de la estructura de la base de datos, cada tabla tiene una llave primaria, un identificador único, compuesto por una o más columnas.

Para establecer la relación entre dos tablas es necesario incluir, en forma de columna, en una de ellas la llave primaria de la otra. A esta columna se le llamara llave secundaria y esta permitirá expresar las relaciones entre los objetos.

#### *Características de una base de datos relacional*

En secciones anteriores se definió la cardinalidad y características de las bases de datos por lo que se retoma algunos conceptos.

- *Cardinalidad en una base de datos relacional.*

Los números mínimo y máximo de veces que una relación puede producirse entre ocurrencias de dos entidades se conocen como cardinalidad. Los tipos de relaciones que se pueden presentar son:

- Relación uno a uno, también denominada relación de matrimonio, consiste en relacionar una entidad A con una entidad B y viceversa

- Relación uno a muchos , la entidad A puede relacionarse con cualquier entidad del tipo B, pero B o las de su tipo solo pueden relacionarse con la entidad A
  - Relación muchos a uno, indica que la entidad B puede relacionarse con cualquiera de las entidades del tipo A, mientras que cada entidad del tipo A solo puede relacionarse con solo una entidad del tipo B
  - Relación muchos a muchos, establece cualquier cantidad de entidades del tipo A pueden estar relacionados cualquier cantidad de entidades del tipo B
- *Integridad referencial en una base de datos relacional*

Se entiende por integridad referencial a las reglas que se establecen para mantener las relaciones entre las tablas cuando se agregan, cambian o eliminan registros. Al exigir la integridad referencial, se impide a los usuarios que agreguen registros a la tabla relacionada para la cual no hay llave principal, cambiar los valores de la tabla principal que darían lugar a registros huérfanos en la tabla relacionada, así como eliminar registros de una tabla principal cuando no hay registros relacionados coincidentes.

Los términos formales del modelo relacional a menudo son sustituidos por otros de uso común, debido a que estos términos son demasiado abstractos para ser usados en la práctica.

<b>Término relacional formal</b>	<b>Equivalente informal</b>
<b>Relación</b>	Tabla
<b>Tupla</b>	Fila o registro
<b>Cardinalidad</b>	Número de filas o registros
<b>Atributo</b>	Columna o campo

<b>Grado</b>	Número de columnas o campos
<b>Llave primaria</b>	Identificador único
<b>Dominio</b>	Conjunto de valores permitidos para el atributo

*Tabla 8. Términos del modelo relacional.*

En el entorno actual de desarrollo de sistemas es importante considerar un buen diseño de base de datos que nos ayuden a reflejar la estructura del problema y que sea capaz de representar todos los datos esperados, en este sentido se recomienda un diagrama Entidad-Relación, ya que por medio de este es posible determinar como el modelo relacional encaja, y que atributos son llaves primarias y cuales llaves secundarias.

El resultado de este proceso es una base de datos normalizada que facilita el acceso a los datos y evita duplicidad. El diseño formal de una base de datos se centra en asegurar que el diseño se ajuste a un nivel de normalización. La idea es construir un modelo lógico que cumpla con las reglas de normalización de datos. La aplicación de una de estas reglas es de una operación que toma una relación como argumento de entrada y da como resultado dos o más relaciones que cumplen con lo siguiente:

- La relación a la que se le aplica la regla, es desestimada en el nuevo esquema relacional
- No se introducen nuevos atributos en el esquema relacional que resulta de la normalización
- Los atributos de la relación a la que se le aplica la regla de normalización, pasan a formar parte de una o más relaciones resultantes

Al hablar de una base de datos relacional es necesario conocer el concepto de *normalización* mismo que según (Juárez, 2006, p. 53) es una técnica que se utiliza para comprobar la validez de los esquemas lógicos basados en el modelo relacional, ya que asegura que las relaciones (tablas) obtenidas no tengan datos redundantes.

La normalización se utiliza para mejorar el esquema lógico y garantiza que el esquema resultante se encuentre lo más próximo al flujo de información con mínima redundancia y máxima estabilidad. Este proceso de normalización

Según (Martínez, 2006, p. 49) existen varias reglas de normalización, sin embargo, es suficiente con garantizar que se cumplan las tres primeras formas normales.

- Primera forma normal (FN1)

Una relación R satisface la primera forma normal si cumple las condiciones de una relación que cumpla:

- ✓ En las celdas de la tabla bidimensional debe haber valores individuales
- ✓ Todas las entradas en cualquier columna deben ser del mismo tipo
- ✓ Cada columna debe tener un nombre único y no importa el orden de las columnas en la tabla
- ✓ No puede haber dos renglones idénticos
- ✓ Segunda forma normal (FN2)

Una relación R satisface la segunda forma normal si, y solo si, satisface las reglas de FN1 y todos sus atributos no-llave dependen por completo del atributo llave.

- ✓ Tercera forma normal (FN3)

Una relación R está en tercera forma normal si está en la FN2 y no tiene dependencias transitivas.

El modelo E-R es de fundamental importancia para el diseño de una base de datos relacional por tanto es necesario conocer que para el correcto diseño del diagrama, se debe realizar lo siguiente:

- ✓ Identificar las entidades que debe presentar la base de datos
- ✓ Determinar las cardinalidades de las interrelaciones establecidas entre las distintas entidades y clasificar estas interrelaciones entre los siguientes tipos:  
uno a uno, uno a muchos, muchos a uno o muchos a muchos
- ✓ Dibujar el diagrama E-R

- ✓ Determinar los atributos de cada entidad

Por otro lado, para transformar el diagrama E-R al diseño de la base de datos, se realiza el siguiente procedimiento:

- ✓ Las entidades en las que hay una relación uno a uno se deben fusionar en una sola entidad
- ✓ Cada una de las entidades que queden se convierte en una tabla con llave primaria y una serie de atributos, de los cuales algunos pueden ser llaves secundarias
- ✓ Las interrelaciones uno a muchos se transforman en atributo y llave secundaria de la tabla que representa a la entidad situada del lado de la interrelación correspondiente a muchos
- ✓ Las interrelaciones muchos a muchos entre dos entidades pasan a ser una tercera tabla con llaves procedentes de ambas entidades. Estas llaves secundarias deberán formar parte de la llave primaria de la tabla en que se convierte la interrelación, no se entiende cuando corresponda.

En resumen (Navarro, 2009, p.26) menciona que el diseño de una base de datos relacional consiste en primera instancia en planificar el tipo de información disponible y la información que necesitamos. Después se procede a diseñar la estructura de una tabla misma que consiste en la descripción de los campos, registros y valores que la componen.

### ***Ventajas de las bases de datos relacionales***

Las bases de datos relacionales presentan numerosas ventajas:

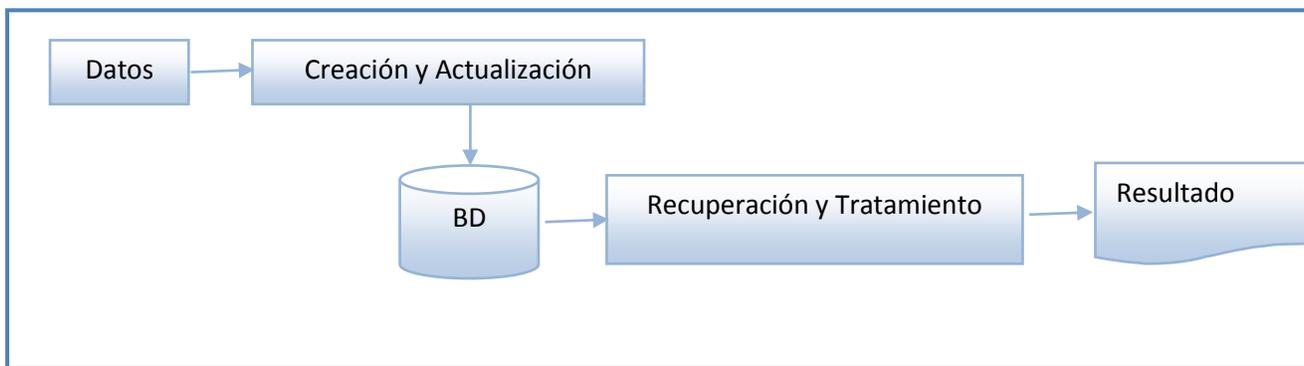
- Controlar la redundancia de datos, no se almacenan varias copias del mismo dato
- Consistencia de datos, controlando la redundancia de datos se reduce en gran medida el riesgo de que haya inconsistencia
- Concurrencia de datos, la base de datos puede ser utilizada por todos los usuarios autorizados en forma simultanea
- Mantener estándares, facilita el intercambio de datos
- Integridad de datos, no se pueden violar las reglas establecidas conforme a restricciones de los datos y sus relaciones.

### 3.4 Manejador de bases de datos

Según (Burbano, 2006, p.17) es un conjunto de programas interrelacionados que se encargan de manejar la creación y todos los accesos a las bases de datos.

El objetivo principal de un manejador de bases de datos es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea práctica y eficiente. El DBMS (Data Base Management System) es conocido también como Gestor de Base de datos.

En la siguiente figura se aprecia la interacción del usuario de la base de datos.



*Figura 23. Diagrama*

En sí, un manejador de base de datos es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar. Un manejador de bases de datos ayuda a crear y organizar una base de datos de tal manera que permita acceder a la información que contiene ésta eficientemente.

De igual manera permite interactuar con sentencias en DML al comando del sistema para el manejo de la información. También debe respaldar y recuperar fácilmente la información en caso de fallas en el sistema.

Es importante mantener la seguridad e integridad de la base de datos por lo que se debe tener un estricto control de cambios en el manejador de bases de datos. Por lo cual se debe dar permisos a usuarios y grupos de usuarios.

La consistencia de datos es muy importante por lo que debe mantenerse siempre y en aquellos casos en los que no se ha logrado eliminar inconsistencias, se debe tener cuidado con la información que sigue repetida, y que ésta se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

Debe manejar transacciones, que es un programa que se ejecuta como una sola operación, es decir, luego de una ejecución fallida la base de datos queda igual como si el programa no se hubiera ejecutado.

Debe mantener varios niveles de abstracción al usuario, es decir, no importa si la base de datos está compuesta por uno o muchos archivos, el usuario debe percibir el sistema igual. Se puede decir que el sistema ahorra a los usuarios detalles del almacenamiento físico de la información.

Debe mantener independencia de datos, es decir, tener la capacidad de modificar el esquema de una base de datos sin modificar las aplicaciones en ella.

El tiempo de respuesta debe ser el mínimo en darnos la información solicitada y en almacenar los cambios realizados.

Según (Elmasri, 2004, p.46) existen utilidades en el manejador que ayudan a la administración de la base de datos, estos son:

1. Carga (Loading): Se utiliza para cargar los archivos de datos existentes, tales como archivos de texto o archivos secuenciales, en la base de datos. Por lo general, el formato de los datos de origen y destino son previamente especificados en la utilidad, la cual formatea automáticamente los datos y los almacena en la base de datos. Con la proliferación de DBMS transferir datos de un manejador a otro es cada vez más común en muchas organizaciones. Estas herramientas se llaman también herramientas de conversión.
2. Copia de seguridad (Backup): La copia de seguridad respalda toda la base de datos, por lo general, en una cinta. La copia de seguridad puede ser usada para restaurar la base de datos en caso de falla catastrófica. Las copias de seguridad incrementales

son también de uso frecuente, donde se registran los cambios sólo desde la copia de seguridad anterior aunque es más complejo, pero ahorra espacio.

3. Reorganización de archivos: Esta utilidad permite reorganizar archivos de base de datos para mejorar el rendimiento.
4. Monitoreo del desempeño: Esta utilidad controla el uso de bases de datos y proporciona estadísticas al administrador. Las estadísticas sirven para la toma de decisiones tales como reorganizar o no los archivos para mejorar el rendimiento.

En la siguiente tabla se muestra la comparativa de los DBMS más comunes.

Características	Oracle	SQL Server	Informix	MySQL
<b>Plataforma</b>	Windows, Linux, Unix, Mac OS X, BSD	Windows	Windows, Linux, Unix, Mac OS X, BSD	Windows, Linux, Unix, Mac OS X, BSD
<b>ACID</b>	SI	SI	SI	SI
<b>Triggers</b>	SI	SI	SI	SI
<b>Transacciones Concurrentes</b>	SI	SI	SI	SI
<b>Funciones, cursores, procedimientos</b>	SI	SI	SI	SI
<b>SQL ANSI 99</b>	SI	SI	SI	SI
<b>SQL ANSI 92</b>	SI	SI	SI	SI
<b>Número de Usuarios</b>	SI	SI	SI	SI

<b>Ilimitados</b>				
<b>Integridad Referencial</b>	SI	SI	SI	SI
<b>Transacciones</b>	SI	SI	SI	SI
<b>ODBC</b>	SI	SI	SI	SI
<b>Licencia de Software</b>	Propietario	Propietario	Propietario	GPL
<b>Cliente/Servidor</b>	SI	SI	SI	SI
<b>MVCC</b>	SI	SI	SI	SI
<b>Costo</b>	Alto	Medio	Alto	Ninguno

*Tabla 9. Características de los manejadores de BD más comunes frecuentes.*

Para la posterior implementación de la tesis, se considerará al manejador de bases de datos MySQL ya que es libre de costo y tiene licencia GPL a diferencia de los otros que se muestran en la tabla comparativa, además porque permite la comunicación cliente servidor, uso de diferentes plataformas, entre otras características que se consideran importantes.

A continuación se abordará sobre el tema de Inteligencia Artificial.