

7



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN**

**“BASES DE DATOS: BASE DE DATOS
PARA LA ADMINISTRACION Y CONTROL
EN EL ACCESO EN LOS SISTEMAS
JURIDICOS”**

TRABAJO DE SEMINARIO

**QUE PARA OBTENER EL TITULO DE
LICENCIADO EN INFORMATICA
P R E S E N T A**

RENE RUBEN LAZCANO LOPEZ

ASESOR:

ING. VICTOR HUGO ARROYO HERNANDEZ

CUAUTITLAN IZCALLI, EDO. DE MEX. 2000

287300



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES



UNIVERSIDAD NACIONAL
SISTEMA DE
MEXICO

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLAN



DEPARTAMENTO DE
EXAMENES PROFESIONALES

DR. JUAN ANTONIO MONTARAZ CRESPO
DIRECTOR DE LA FES CUAUTITLAN
PRESENTE

ATN: Q. Ma. del Carmen García Mijares
Jefe del Departamento de Exámenes
Profesionales de la FES Cuautitlán

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario:

Bases de Datos: Base de Datos para la Administración y Control en el Acceso a los Sistemas

Jurídicos

que presenta el pasante: René Rubén Lazcano López

con número de cuenta: 8928922-4 para obtener el título de
Licenciado en Informática

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXÁMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán Izcalli, Méx. a 24 de Octubre de 2000

MODULO	PROFESOR	FIRMA
I	Ing. Víctor Hugo Arroyo Hernández	
II	MC. Araceli Nivon Zaghi	
IV	Lic. Carlos Pineda Muñoz	

*A esa gran fuerza que toda la humanidad venera,
con diferentes nombres en cada lugar,
pero con la misma grandeza.*

*A mis padres:
Por darme la vida, la filosofía, el valor al resolver cualquier conflicto,
Al sacrificio, al ánimo y a la fuerza de voluntad de mi Padre,
El entusiasmo, el empuje, la fe y energía de mi Madre.*

*A mis hermanas:
Gracias al gran aliento y absoluto amor que siempre me han dado,
A esa gran fuerza, que irradian juntos.*

*A toda mi familia:
Por el empuje que todos tienen,
ese gran apoyo en las adversidades que me han brindado
y los muy buenos momentos que paso a paso hemos vivido.*

*A mi pareja:
Por su inspiración, estímulo y comprensión,
Pero sobre todo por su gran corazón.*

*A todos:
Gracias por su ahínco y Amor.*

INDICE

I.- HISTORIA DEL ARTE DE LAS BASES DE DATOS	1
Definición de Base de Datos	4
Arquitectura de Bases de Datos	4
Nivel Físico o Interno	5
Nivel Conceptual	5
Nivel Externo o de Vistas	5
Sistema de Gestión de Bases de Datos (SGBD)	6
Arquitectura	7
Funciones	8
Ventajas e inconvenientes	8
Componentes	10
Lenguaje de Definición de Datos (DDL Data Definition Language)	10
Lenguaje de Manipulación de Datos (DML Data Manipulation Language)	10
II.- MODELOS DE DATOS y MODELO RELACIONAL	12
Modelos Conceptuales	14
Modelos Lógicos	16
Modelo Jerárquico	17
Modelo de Red	18
Modelo Relacional	19
Modelo Relacional	20
Historia	20
Conceptos fundamentales del Modelo Relacional	20
Restricciones de Integridad	22
a) Regla de Integridad de la Entidad	22
b) Regla de Integridad referencial	22
c) Dependencias Funcionales	23
Normalización	23

INDICE

III.- LENGUAJE DE CONSULTA ESTRUCTURADO (SQL) y ARQUITECTURA CLIENTE / SERVIDOR.	26
Álgebra relacional	26
Cálculo relacional	30
Lenguaje de consulta SQL	31
Comandos y Descripción	32
Creación de tablas	32
Borrado de tablas	32
Modificación de la estructura de las tablas	32
Introducción o inserción de tuplas	33
Consulta	34
Combinación de consultas	38
Permisos sobre tablas	39
Vistas	39
Cliente / Servidor	41
Middleware	45
IV.- CASO PRACTICO. BASE DE DATOS PARA LA ADMINISTRACIÓN Y CONTROL EN EL ACCESO EN LOS SISTEMAS JURÍDICOS.	
Planteamiento del problema	47
Requerimientos	48
Ambiente de Trabajo	49
Solución al Proyecto	49
Definición de la base de datos	50
Sistema Prototipo	59
Conclusiones	78
BLIBLIOGRAFIA.	
GLOSARIO.	

INDICE DE FIGURAS

Cap. I.	Figura 1.	Arquitectura General del Sistema de Gestión de Bases de Datos	7
Cap. I.	Figura 2.	Lenguajes de Programación para las Bases de Datos	11
Cap. II.	Figura 3.	Modelo de Datos.	13
Cap. II.	Figura 4.	Modelo Conceptual.	14
Cap. II.	Figura 5.	Modelo Jerárquico	17
Cap. II.	Figura 6.	Modelo de Red.	18
Cap. II.	Figura 7.	Modelo Relacional.	19
Cap. II.	Figura 8.	Formas Normales.	24
Cap. III.	Figura 9.	El Cliente	42
Cap. III.	Figura 10.	El Servidor	43
Cap. III.	Figura 11.	Base de Datos, Proceso y Presentación Arquitectura Cliente/Servidor	43
Cap. III.	Figura 12.	Procesos Arquitectura Cliente/Servidor	44

INTRODUCCIÓN

Hoy en día existe un problema fundamental en la impartición de justicia, que es la confiabilidad. Una gran dificultad cuesta saber si la información con que se esta examinando es fiable, exacta y ante todo que no ha sido modificada en ninguno de sus puntos.

Para proteger de fugas de información, se pretende tener un control más exacto de todos los integrantes del sistema de impartición de justicia, y para fines de este trabajo se implementará un sistema para resguardar una parte de todo ese Sector de Justicia.

En el área de **Seguridad de Datos** de la División de Justicia, existe la necesidad de asegurar la información sensible de todos los Usuarios existentes en los diferentes Sistemas Penales de la Institución; por lo mismo es de gran preocupación por la relevancia de la información que se maneja, y habiendo el problema de fugas de información, es como nace la idea de generar un programa que pueda administrar, controlar y en general tener un control pleno del personal que se encuentra dado de Alta en cada Sistema y además a qué recursos tiene acceso.

Con la implementación de esta de bases de datos se registrará al personal que maneja el sistema y se tendrá un control pleno de los mismos así como de la información que manejan las Áreas Penales, actualizando todos y cada uno de los datos, esto es, una administración completa tanto de los **Usuarios** como a los **Cuatro Sistemas** que Administran, dando como resultado un confiable sistema de impartición de Justicia.

OBJETIVO GENERAL

Incrementar la fiabilidad de los sistemas penales, al llevar el control y administración mediante una base de datos de un Sistema de Seguridad de Datos para solucionar el problema de manejo de los sistemas penales y concluir con la fuga de información que existe.

OBJETIVO PARTICULAR

Implementar un **Sistema de Control de Usuarios en Seguridad de Datos**, a fin de administrar, vigilar y definir los diferentes perfiles y niveles de acceso, que tengan los Usuarios en los diferentes cuatro Sistemas Penales, para garantizar la seguridad en el uso de transacciones sensibles y evitar daños a la institución.



CAPITULO I

HISTORIA DEL ARTE DE LAS BASES DE DATOS.

- ◆ Definición de Base de Datos.
- ◆ Arquitectura de Bases de Datos.
- ◆ Sistema de Gestión de Bases de Datos.
- ◆ Lenguaje de Definición de Datos.
- ◆ Lenguaje de Manipulación de Datos.

HISTORIA DEL ARTE DE LAS BASES DE DATOS.

La sofisticación de la tecnología de las bases de datos es el resultado de la evolución que a lo largo de varias décadas ha tenido lugar en el procesamiento de los datos y en el manejo de la información.

A la vez que las empresas aumentaban su dependencia respecto a los ordenadores durante los años 60, los requisitos para la gestión de datos se tornaron críticos.

Los primeros sistemas de procesamiento de datos ejecutaron las tareas administrativas para reducir el papeleo. Los sistemas computacionales se utilizaron inicialmente en los negocios para funciones de contabilidad, las cuales eran imprescindibles para que funcionara el negocio, estos sistemas se llamaron **sistemas de procesamiento de datos** ya que ejecutaban las funciones habituales de tratamiento de los registros de la organización. Este sistema de procesamiento de archivos era secuencial es decir, que cada registro puede leerse y procesarse únicamente después que todos los registros que lo proceden en el archivo hubieran sido leídos.

Las limitaciones de los sistemas orientados a archivos puramente secuenciales no los privaron de ser herramientas eficaces para realizar las transacciones en las empresas, sin embargo, para ejecutar muchas tareas rutinarias en los negocios se necesitaba el **acceso directo a los datos** es decir, la capacidad de tener acceso y procesar directamente un registro dado sin ordenar primero el archivo o leer los registros en secuencia, sin embargo, los archivos de acceso directo solamente proporcionaron una solución parcial. Para lograr una solución más completa a estos problemas fue necesario introducir los sistemas de gestión de bases de datos.

Un sistema de bases de datos está formado por:

- ✓ Una base de datos,
- ✓ Un sistema computacional de propósito general (llamado Sistema de Gestión de Bases de Datos SGBD), que manipula la base de datos, y;
- ✓ Hardware y personal apropiados.

A pesar de la introducción de los archivos de acceso directo, pronto se hizo obvio que a los sistemas de archivo de cualquier tipo era inherente un conjunto de deficiencias:

1. **Redundancia de los datos.** Algunos elementos de los datos eran comunes a varias aplicaciones.
2. **Pobre control de los datos.** El mismo elemento de los datos solía tener varios nombres.
3. **Capacidades inadecuadas de manipulación de los datos.**
4. **Esfuerzo excesivo de programación.**

Los sistemas de bases de datos superan estas limitaciones de los sistemas orientados a los archivos. Al tolerar una estructura de datos centralizada, integrada, los sistemas de base de datos eliminan los problemas de redundancia y de control de los datos.

En la actualidad, estamos inmersos en varias décadas de largo esfuerzo por desarrollar sistemas de gestión de bases de datos más poderosos. El desarrollo de los sistemas se ha dado en tres modelos de datos fundamentales, y que no son más que métodos conceptuales para estructurar los datos. Estos tres modelos de datos son el **jerárquico, en red y relacional**.

Los primeros sistemas de bases de datos, introducidos a mediados de los años 70, estaban basados en el **modelo jerárquico**, que presume que todas las interrelaciones entre los datos pueden estructurarse como jerarquías, es decir, los archivos se conectan entre sí mediante punteros (apuntadores) físicos o campos de datos añadidos a los registros individuales.

Un puntero (apuntador) es una dirección física que identifica dónde puede encontrarse un registro sobre el disco. Sin embargo, se comprobó muy rápidamente que este modelo tenía algunas limitaciones importantes, ya que no todas las interrelaciones podrían expresarse fácilmente en una estructura jerárquica.

A finales de los 70 se desarrollaron los sistemas de base de datos en red, los cuales también emplearon punteros físicos para enlazar entre sí los registros de diferentes archivos.

El uso de punteros tenía ciertas ventajas y desventajas, era de gran ayuda porque permitieron la recuperación rápida de los datos que tuvieran interrelaciones predeterminadas; la debilidad está en el hecho de que estas interrelaciones tenían que definirse antes de que el sistema fuera puesto en explotación. Era difícil recuperar datos basados en otras interrelaciones.

En 1970 Edgar F. Codd propuso un modelo simple de datos en el que todos ellos se representarían en tablas constituidas por *filas* y *columnas*. A estas tablas se les dio el nombre matemático de relaciones, de ahí nació el **modelo relacional**. De igual forma Codd propuso dos lenguajes para manipular los datos en las tablas: el **Álgebra Relacional** y el **Cálculo Relacional**, ambos lenguajes soportan la manipulación de los datos sobre la base de operadores lógicos en lugar de los punteros físicos utilizados en los modelos jerárquico y de red.

Todos estos desarrollos hicieron avanzar enormemente el estado del arte en los sistemas de gestión de bases de datos y aumentaron la disponibilidad de información en las bases de datos colectivas.

Actualmente, los sistemas relacionales son un estándar en el mercado, especialmente en operaciones comerciales.

Ventajas de las Bases de Datos.

La diferencia de una Base de Datos respecto a otro sistema de almacenamiento de datos es que éstos se almacenan de forma que cumplan tres requisitos básicos:

- No redundancia:** Los datos se almacenan una sola vez. Si varias aplicaciones necesitan los mismos datos no crearán cada una su propia copia sino que todas accederán a la misma.
- Independencia:** Los datos se almacenan teniendo en cuenta la estructura inherente a los propios datos y no la de la aplicación que los crea. Esta forma

de trabajar es la que permite que varias aplicaciones puedan utilizar los mismos datos. Se puede hablar de dos tipos de independencia: **independencia física**, de tal manera que la estructura física de la Base de Datos puede ser modificada de forma transparente para los programas que la utilizan, e **independencia lógica**, es decir el programador usa la Base de Datos pero desconoce su estructura interna.

Concurrencia: Varios usuarios, ejecutando la misma o diferente aplicación, podrán acceder simultáneamente a los datos.

DEFINICIÓN DE BASE DE DATOS.

Una **base de datos** en general, es un conjunto de datos no redundantes, almacenados en un soporte informático, organizados de forma independiente de su utilización y accesibles simultáneamente por distintos usuarios y aplicaciones.

Es una colección de elementos de datos interrelacionados que pueden procesarse por uno o más sistemas de aplicación.

ARQUITECTURA DE BASES DE DATOS.

En las bases de datos una arquitectura aceptada internacionalmente considera tres niveles:

1. **Un nivel físico.**
2. **Un nivel conceptual.**
3. **Un nivel lógico.**

Con esta arquitectura en las bases de datos, se pretende hacer una separación entre las aplicaciones y la base de datos física. Además puede obtener ciertas características como independencia de los datos y aplicaciones, y las vistas que puede obtener el usuario.

Nivel Físico o Interno.

Nivel más bajo de abstracción de datos. Se especifica la forma en que se organizan los datos físicamente, es decir, describe la estructura física de almacenamiento de la base de datos, describe los detalles para el almacenamiento de los mismos y los caminos de acceso que van a existir para la base de datos.

Incluye la asignación de espacios para cada conjunto de datos, en los caminos de acceso se especifican las claves, índices y punteros.

Nivel Conceptual.

Describe la estructura de toda la base de datos para los usuarios, incluye la descripción de todos los datos así como las relaciones que hay entre éstos y en donde se definen las restricciones de integridad y de confidencialidad.

Se ocultan los detalles de las estructuras físicas de almacenamiento y se describen las entidades, los tipos de datos, vínculos, operaciones de los usuarios y restricciones.

Nivel Externo o de Vistas.

También llamado nivel de vistas debido a que describe la parte de la base de datos que le interese a un grupo de usuarios determinado, ocultando el resto de la base de datos a ese grupo, sólo describe una parte de la base de datos.

Estos tres niveles sólo contienen descripciones de los datos, existiendo datos realmente en el nivel interno.

Los sistemas gestores de bases de datos no llegan a distinguir los tres niveles de abstracción pero algunos de ellos se basan en este tipo de arquitectura y deben ofrecer lenguajes para definir la base de datos conceptual, las vistas externas y la representación almacenada de la base de datos.

Inicialmente, los servicios de datos se incorporaban directamente en las aplicaciones y utilizaban el sistema de archivos del sistema operativo. Esto condujo inevitablemente a archivos de datos que eran incompatibles con otras aplicaciones, cuyo mantenimiento y actualización eran difíciles. Cambiar las estructuras de datos implicaba frecuentemente cambiar también la aplicación. Desde el desarrollo de los **Sistemas de Gestión de Bases de Datos (SGBD)**, las aplicaciones no necesitaron incluir estos servicios de datos.

SISTEMA DE GESTIÓN DE BASES DE DATOS (SGBD).

Un sistema de gestión de bases de datos (SGBD¹) es un software, parecido a un sistema operativo o a un compilador, que brinda un conjunto de servicios a los usuarios finales, los programadores y otros; en si un **Sistema de Gestión de Bases de Datos (SGBD)** es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una Base de Datos nunca se accede o manipula directamente sino a través del Sistema de Gestión de Bases de Datos. Se puede considerar al Sistema de Gestión de Bases de Datos como el interfaz entre el Usuario y la Base de Datos.

El funcionamiento del Sistema de Gestión de Bases de Datos está muy interrelacionado con el del Sistema Operativo², especialmente con el sistema de comunicaciones. El Sistema de Gestión de Bases de Datos utilizará las facilidades del sistema de comunicaciones para recibir las peticiones del usuario (que puede estar utilizando un terminal físicamente remoto) y para devolverle los resultados.

Un Sistema de Gestión de Bases de Datos cuenta con las características siguientes:

- *Seguridad e integridad de la información de una Base de Datos.*
- *Consulta de la información mediante el uso de lenguajes de consulta y escritores de reportes.*

¹ También se puede hacer referencia a los sistemas de gestión de bases de datos, por su nombre en Inglés Data Base Management System (DBMS)

² Conjunto de programas responsable de la explotación del ordenador, mejorar su nivel de rendimiento y gestionar los recursos del mismo.

- ❶ *Entrada de datos y actualización interactiva.*
- ❷ *Independencia de los datos.*

Arquitectura de un Sistema de Gestión de Bases de Datos.

La mayoría de los Sistemas de Gestión de Bases de Datos actuales están inspirados en una arquitectura sugerida en 1978 por un grupo de trabajo de ANSI³ es una arquitectura adecuada para construir Bases de Datos que cumplan los requisitos señalados en la definición y además sean portables entre distintas máquinas y sistemas operativos.

La figura 1 muestra como esta arquitectura divide la base de datos en los tres niveles que anteriormente se explicaron como son:

- El nivel Externo ó de Vistas
- El nivel Conceptual
- El nivel Interno ó Físico

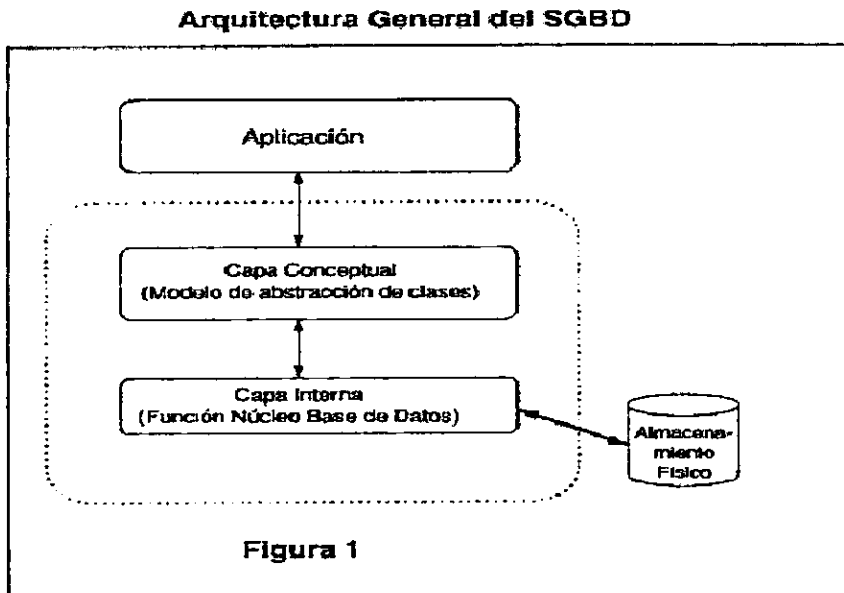


Figura 1.

³ ANSI American National Standards Institute. Instituto Nacional Americano para la Estandarización. Organismo oficial dedicado a fomentar la adopción de normativas en materia de informática, comunicaciones, etc.

Funciones de un Sistema de Gestión de Bases de Datos.

Un Sistema de Gestión de Bases de Datos debe proporcionar un amplio surtido de funcionalidades para poder cumplir adecuadamente su misión. Normalmente se clasifican en definición, manipulación y utilización.

- **Función de definición:** Permite describir los elementos de datos, sus estructuras, sus interrelaciones y sus validaciones a nivel externo, lógico e interno. Esta función es realizada por una parte del Sistema de Gestión de Bases de Datos denominada Lenguaje de Definición de Datos (LDD o DDL, *Data Definition Language*).
- **Función de manipulación:** Permite buscar, añadir, suprimir y modificar los datos de la Base de Datos. Esta función es realizada por una parte del Sistema de Gestión de Bases de Datos denominada Lenguaje de Manipulación de Datos (LMD o DML, *Data Manipulation Language*).
- **Función de utilización:** Incluye otras funcionalidades tales como: modificar la capacidad de los registros, cargar archivos, realizar copias de seguridad, arranque, protección frente a accesos no autorizados, gestión de la concurrencia, estadísticas de utilización, etc.

Un Sistema de Gestión de Bases de Datos puede también presentar los datos a una aplicación de una forma que sea muy distinta a las estructuras de datos físicas. La forma en que los datos se presentan se denomina abstracción de datos, y determina el modelo de datos del Sistema de Gestión de Bases de Datos.

Ventajas e inconvenientes de los Sistema de Gestión de Bases de Datos.

La instalación de un Sistema de Gestión de Bases de Datos en un sistema que está funcionando sin él, normalmente proporciona una amplia serie de ventajas. Entre las más importantes se pueden destacar:

- ◆ **Eliminan las inconsistencias en los datos.** Esto es algo especialmente difícil sin un Sistema de Gestión de Bases de Datos cuando los mismos datos se utilizan y actualizan en diferentes procesos.
- ◆ **Permiten compartir los mismos datos** entre diferentes aplicaciones con distintas necesidades. Por ejemplo: aplicaciones transaccionales junto con aplicaciones de soporte a la dirección.
- ◆ **Se adaptan mejor a la existencia de aplicaciones rápidamente cambiantes.** En estos casos con los enfoques tradicionales se puede requerir la conversión de los datos cada vez que se requiera. Un Sistema de Gestión de Bases de Datos proporcionará independencia de los datos respecto a las aplicaciones.
- ◆ **Ahorran espacio de almacenamiento** al no existir redundancia o ser ésta escasa. También porque muchos Sistemas de Gestión de Bases de Datos utilizan mecanismos de compresión para almacenar los datos.
- ◆ **Mejoran la seguridad de los datos** pues normalmente incorporan mecanismos de seguridad en el propio Sistema de Gestión de Bases de Datos.
- ◆ **Permiten la creación de entornos de alta disponibilidad.** Los Sistemas de Gestión de Bases de Datos modernos suelen permitir realizar gran parte (a veces todo) del mantenimiento del sistema sin necesidad de parar las aplicaciones. Por tanto, con algunos Sistemas de Gestión de Bases de Datos es posible llegar a disponer de aplicaciones funcionando ininterrumpidamente.

Por otra parte, si se escoge adecuadamente el Sistema de Gestión de Bases de Datos no suelen presentarse problemas de tipo técnico que no se presenten con los sistemas anteriores de almacenamiento de datos sino que los problemas suelen ser los típicos de cualquier equipo lógico complejo:

- ◆ **La puesta en funcionamiento puede ser larga.** Pues antes de obtener los primeros resultados se necesita un período de formación y adaptación variable según la complejidad del entorno.

- ◆ **Se necesita personal especializado para su mantenimiento.** En principio un Diseñador de la Base de Datos y un Administrador permanente de la Base de Datos.

Componentes de un Sistema de Gestión de Bases de Datos.

Lenguaje de Definición de Datos (DDL: *Data Definition Language*).

Sencillo lenguaje artificial para definir y describir los objetos de la base de datos, su estructura, relaciones y restricciones. En la práctica puede consistir en un subconjunto de instrucciones de otro lenguaje informático. Aparte suele poseer dos subconjuntos de instrucciones:

- ☛ Lenguaje de definición del almacenamiento de los datos (DSDL: *Data Storage Definition Language*): permite especificar características físicas de la base de datos (volúmenes y archivos donde van a ser almacenados los datos, etc).
- ☛ Lenguaje de Control de Datos (DCL: *Data Control Language*): encargado del control y seguridad de los datos (privilegios y modos de acceso, etc).

Lenguaje de Manipulación de Datos (DML: *Data Manipulation Language*).

Lenguaje artificial de cierta complejidad que permite el manejo y procesamiento del contenido de la base de datos. En la práctica puede consistir en un subconjunto de instrucciones de otro lenguaje informático.

Las aplicaciones que trabajan sobre la base de datos se programan en un lenguaje de programación (C⁴, Cobol⁵,...) insertando en el código fuente sentencias del Lenguaje de Manipulación de Datos (*figura 2*). Al utilizar un Lenguaje de Manipulación de Datos se deben especificar los datos que serán afectados por las sentencias del lenguaje. Un

⁴ C. Lenguaje de programación estructurado, de aplicación general.

Lenguaje de Manipulación de Datos puede tener o no procedimientos, según sea necesario especificar cómo deben obtenerse esos datos. Los Lenguajes de Manipulación de Datos con procedimientos tienen sentencias de control de flujo como bucles o condicionales. Los Lenguajes de Manipulación de Datos sin procedimientos son conocidos también como declarativos. Este lenguaje Lenguaje de Manipulación de Datos es empleado en los Sistemas de Gestión de Bases de Datos para generar un diccionario de datos⁶.

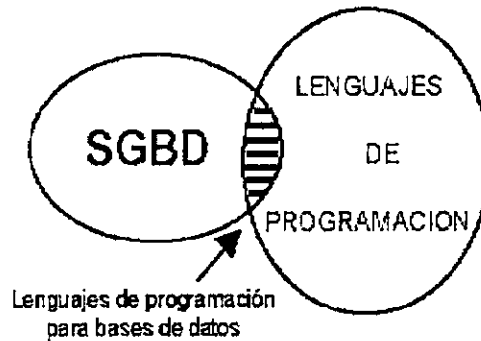


Figura 2.
Lenguajes de Programación para las Bases de Datos

⁵ **COBOL. COmmon Business Oriented Language.** Es un lenguaje de programación que desarrolló CODASYL y, de hecho, es norma en el proceso de datos comerciales.

⁶ Un **diccionario de datos** es una descripción lógica de los datos para el usuario. Reúne la información sobre los datos almacenados en la Base de Datos (descripciones, significado, estructuras, consideraciones de seguridad, edición y uso de las aplicaciones, etc.).



CAPITULO II

MODELO DE DATOS.

- ◆ Modelos Conceptuales.
- ◆ Modelos Lógicos.
- ◆ Modelo Relacional.

MODELOS DE DATOS

Un modelo de datos es un esquema que define cada una de las unidades de información y especifica cómo se relaciona una unidad con otras unidades de información de la Base de Datos.

Dentro de la problemática del diseño de bases de datos, los modelos de datos cumplen un importante rol, pues son las herramientas que nos permiten generar los esquemas de bases de datos y los que regirán su estructura.

El modelo de datos va a definir las reglas por las cuales los datos son estructurados. Esta estructuración, sin embargo, no da una interpretación completa acerca del significado de los datos y de la forma en que serán usados. Las operaciones permitidas sobre datos deben ser definidas.

Los niveles de abstracción de la arquitectura ANSI⁷ facilitan el diseño de una base de datos, al proporcionar instrumentos que ayudan a la estructuración del mundo real hasta llegar a la base de datos física.

En el área de bases de datos existe permanentemente el debate de los modelos de datos (*figura 3*). De acuerdo a la arquitectura de las bases de datos (ANSI/SPARC⁸), podemos distinguir 3 tipos de modelos:

- Modelos externos.
- Modelos conceptuales o global.
- Modelos internos.

⁷ ANSI American National Standards Institute. Instituto Nacional Americano para la Estandarización. Organismo oficial dedicado a fomentar la adopción de normativas en materia de informática, comunicaciones, etc.

⁸ SPARC Standard Planning & Requirement Committee

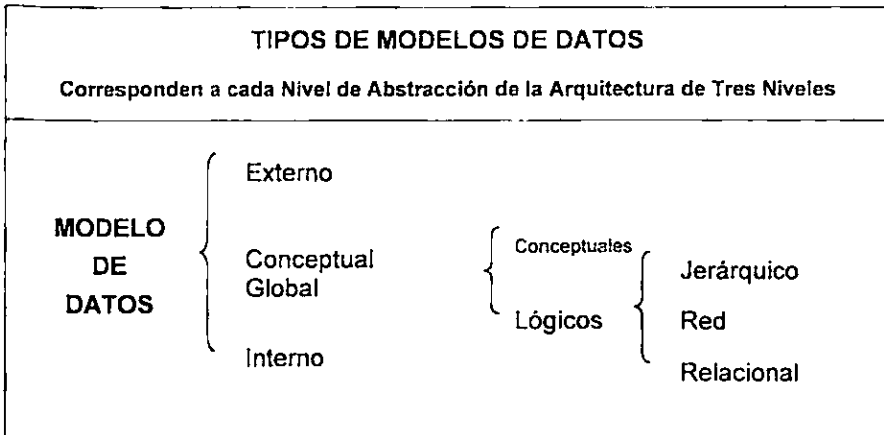


Figura 3. Modelo de Datos.

Los conceptos de un modelo de datos se construyen por lo regular usando mecanismos de abstracción y se describen mediante representaciones lingüísticas y gráficas; es decir, puede definirse una sintaxis y puede desarrollarse una notación gráfica.

Para fines del presente trabajo se analizará la clasificación correspondiente a los modelos **conceptuales** de acuerdo a la arquitectura ANSI/SPARC; de acuerdo a la misma hay dos tipos de modelos de datos: modelos conceptuales, usados en el diseño de bases de datos y modelos lógicos, apoyados por los Sistemas de Gestión de Bases de Datos, que son paquetes de software que crean, modifican y mantienen bases de datos.

Cuando un modelo de datos describe un conjunto de conceptos de una realidad determinada, se llama modelo conceptual de datos.

Los modelos **conceptuales** son instrumentos para representar la realidad a un nivel alto de abstracción. Utilizando los modelos conceptuales, podemos construir una descripción de la realidad fácil de entender e interpretar.

Los modelos **lógicos** apoyan descripciones de datos procesables por un computador, incluyen el modelo jerárquico, de red y relacional.

MODELOS CONCEPTUALES.

En el estado actual de la tecnología, donde aun no existe un modelo conceptual (y su lenguaje de descripción, figura 4) generalizado e independiente del Sistema de Gestión de Bases de Datos, es preferible hacer una distinción entre el esquema conceptual, tal como es concebido en la normativa del Instituto Nacional Americano para la Estandarización (descripción de la organización de acuerdo con un modelo independiente del Sistema de Gestión de Bases de Datos) y lo que es el esquema de los actuales Sistemas de Gestión de Bases de Datos (jerárquica, red, relacional) y su sometimiento a las restricciones que estos impongan.

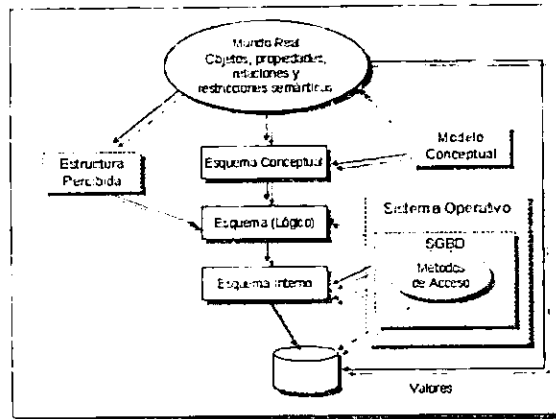


Figura 4.
Modelo Conceptual.

Características.

El diseño de bases de datos no es el único campo para la aplicación de los modelos conceptuales. En los años 80, los llamados sistemas de diccionarios de datos, cuya función es definir el contenido de la base de datos y de los programas de aplicación dentro de los Sistemas de Información, utilizaron extensamente los modelos conceptuales. Principalmente debido a la facilidad de lectura y la expresividad de éstos (documentación de la base de datos).

Una condición básica de los modelos conceptuales es que son buenas herramientas para representar la realidad.

- **Expresividad:** Hacen uso de la abstracción de generalización, de modo que permiten una representación directa en el esquema de una gran variedad de restricciones de integridad, es decir, aserciones que permiten la selección de casos válidos del esquema de base de datos.
- **Simplicidad:** Debe ser simple, para que un esquema creado con este modelo sea fácil de entender por diseñadores y usuarios de la aplicación de bases de datos. Notar que un modelo muy expresivo tiende a ser complejo.
- **Minimalidad:** Si cada concepto presente en el modelo tiene un significado distinto con respecto a todos los demás.
- **Formalidad:** Los esquemas creados usando modelos conceptuales representan una especificación formal. Todos los conceptos del modelo tienen una interpretación única, precisa y bien definida.

MODELOS LÓGICOS.

La organización de la información de una Base de Datos es de acuerdo con un **modelo de datos**

Los modelos lógicos basados en registros se usan para describir los datos en forma lógica y global, siendo de un nivel más alto que el de implementación.

Dentro de la clasificación de modelos lógicos existen diferentes tipos de modelos de datos, algunos de ellos son el jerárquico, de red y el relacional. Siendo el modelo relacional el más utilizado por su rapidez de búsqueda en diferentes campos de la Base de Datos.

Los tres modelos en los que se ha basado el desarrollo de las bases de datos son el jerárquico, en red y relacional. El modelo jerárquico dominó el mercado de los Sistemas de Gestión de Bases de Datos hasta mediados de los 80. Durante este mismo período, surgió el modelo en red con el que se pretendía sustituir a los Sistemas de Gestión de Bases de Datos Jerárquicos, lo que no se consiguió.

A principios de los ochenta el modelo jerárquico comenzó a ser sustituido por una nueva generación de Sistema de Gestión de Bases de Datos basados en el modelo relacional que, en la actualidad, dominan ampliamente el mercado. De esta manera, los modelos jerárquico y de red han caído en desuso, si bien siguen siendo empleados en aquellos sistemas donde se ha conseguido obtener un elevado rendimiento y no nacen nuevas necesidades en el Sistema de Información.

Modelo Jerárquico.

Los **Sistemas de Gestión de Bases de Datos Jerárquicos** fueron los primeros en aparecer. Una base de datos jerárquica se puede visualizar como una estructura en árbol⁹. Las bases de datos jerárquicas son bastante rígidas (figura 5). Una vez diseñada la base de datos, es complejo cambiarla y, además, es necesario un conocimiento amplio de la forma en la que se han almacenado los datos para poder recuperarlos de forma efectiva. Por ello, a pesar de dominar el mercado de Sistemas de Gestión de Bases de Datos en sus comienzos, han ido decayendo y actualmente no se encuentran en el mercado.

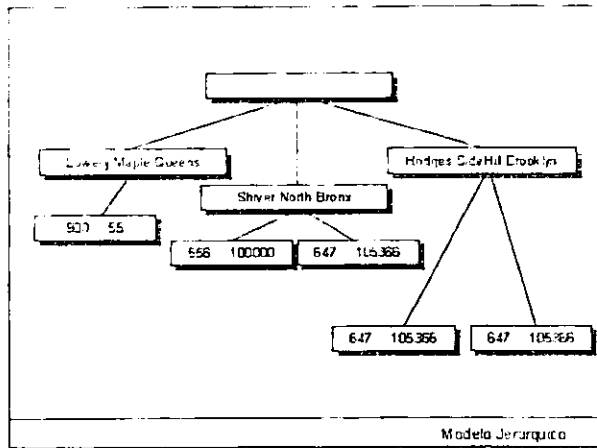


Figura 5.

Por medio de un modelo jerárquico, el esquema de datos puede visualizarse como un árbol, en que los nodos corresponden a las clases de objetos y los arcos corresponden a asociaciones entre nodos.

Este modelo es muy similar al modelo de datos en red, pero con la salvedad de que los registros se organizan con estructura de árbol.

⁹ Estructura de representación de la información que consiste en un único registro "padre" del que dependen cero o más registros "hijos" que, a su vez, pueden dar origen a nuevos subárboles.

Modelo de Red.

También denominado modelo CODASYL¹⁰. Fue el primero en aparecer comercialmente, a principios de los años 70. Los **Sistemas de Gestión de Bases de Datos en red** fueron una evolución del modelo jerárquico.

En una base de datos en red, cada uno de los registros están enlazados entre sí, pero, no necesariamente siguiendo una estructura en árbol. Se caracteriza por almacenar direcciones de otros datos junto a la misma información (figura 6)

El modelo en red elimina parte de las rigideces del modelo jerárquico pero aumenta la complejidad para modificar la estructura de la base de datos. Por ello, a pesar de su buen rendimiento, el número de instalaciones con Sistemas de Gestión de Bases de Datos en red siempre ha sido pequeño y, hoy en día, tampoco se encuentran en el mercado. Sin embargo, aún quedan instalaciones basadas en estos dos modelos de datos que responden con gran eficiencia y plena satisfacción de sus usuarios.

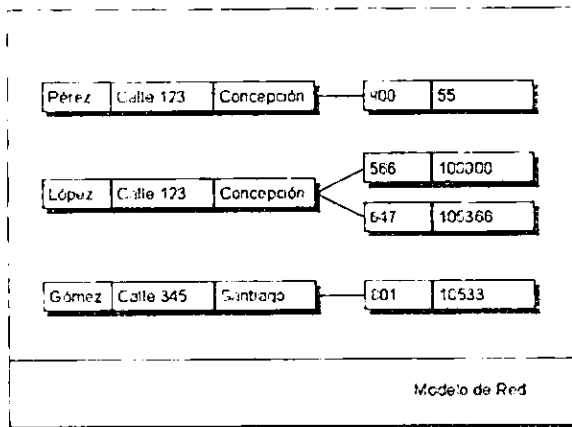


Figura 6.

Los nodos representan clases de objetos
los arcos relaciones entre dos nodos.

¹⁰ *Conference on DATA SYtem Languages*. Organización que fijó el estándar para el modelo de bases de datos en red.

Modelo Relacional.

En una base de datos relacional se representan los datos como un conjunto de tablas bidimensionales compuestas de filas y columnas (*figura 7*). Cada fila representa una relación entre un conjunto de valores y está identificada por una clave única.

Este modelo es el más empleado. Todos los datos visibles al usuario están organizados estrictamente como tablas de valores. Todas las operaciones sobre la base de datos operan sobre esas tablas. Cada fila de una tabla es una instancia de los datos. Cada columna de una tabla es un atributo. Es el modelo de datos más sencillo y cercano a la forma humana de organizar la información.

El modelo relacional organiza la información con estructuras independientes relacionadas entre sí por medio de relaciones lógicas, es decir no unen registros físicamente, pero el diseño de los registros debe proveer un campo común.

Los Sistemas de Gestión de Bases de Datos relacionales son muy flexibles y de fácil manejo, lo que los ha convertido en el modelo dominante en la actualidad.

Un factor decisivo en la implantación de los Sistemas de Gestión de Bases de Datos relacionales ha sido el lenguaje **SQL** (*Structured Query Language*) para la interrogación y el manejo de datos del modelo relacional, unido con la normalización propuesta por Codd para estos tipos de Sistemas de Gestión de Bases de Datos.

Este modelo representa los datos y las relaciones entre los datos mediante una colección de tablas como se muestra a continuación:

Nombre	Calle	Ciudad	Numero
Lowery	Maple	Queens	900
Shaver	North	Bronx	556
Shaver	North	Bronx	647
Hodges	Sidwell	Brooklyn	801
Hodges	Sidwell	Brooklyn	674

Numero	Saldo
900	55.00
556	100 000.00
647	105 366.00
801	10 533.00

Modelo Relacional.

Figura 7.
Modelo Relacional.

MODELO RELACIONAL.

Historia.

La teoría del modelo de datos relacional es obra del investigador de IBM Edgar Codd en 1970. Goza de una fuerte base matemática.

Edgar Codd argumentó que los datos deberían relacionarse mediante **interrelaciones** naturales, lógicas, inherentes a los datos, más que mediante punteros físicos.

El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información debe estar contenida en **tablas**, y las **relaciones** entre datos deben ser representadas explícitamente en esos mismos datos.

En los sistemas de bases de datos relacionales, los archivos completos de datos se pueden procesar con instrucciones sencillas, dicho enfoque mejoró enormemente la eficiencia conceptual de la programación de las bases de datos.

Conceptos fundamentales del Modelo Relacional.

El **modelo de datos relacional** organiza y representa los datos en forma de tablas o relaciones.

En dicho modelo tenemos diferentes términos como el de **Entidades** que son objetos (personas, cosas, lugares, etc.) en el mundo real, se implementan como tablas en una base de datos.

Se componen a su vez de **Atributos** que son las características de interés de las entidades modeladas. Para una persona sus atributos serían Nombre, Apellido, Domicilio, etc. Los atributos corresponden a las columnas en una tabla y el nombre de la columna se llamará por consiguiente nombre del atributo. Se usan los términos atributo y nombre del atributo más que los términos columna y nombre de la columna por ser consistentes con los convenios de bases de datos relacionales.

Una **relación** es término que viene de la matemática y representa una simple tabla de dos dimensiones, consistente en filas y columnas de datos.

El número de atributos de una relación se llama **grado de la relación**.

Las filas de una relación también se llaman **tuplas**. Se asume que no hay un orden preestablecido de las filas o tuplas de la relación y que *dos tuplas no tienen idénticos conjuntos de valores*.

El conjunto de todos los posibles valores que puede tener un atributo es su **dominio**.

Un **valor nulo** no es un espacio en blanco o cero, es simplemente un *valor desconocido* o inaplicable que puede ser remplazado mas tarde, es decir es el valor dado a un atributo en una tupla si el atributo es inaplicable o su valor es desconocido.

Una **clave** es el conjunto mínimo de atributos que identifica unívocamente cada tupla en una relación.

La **clave candidata** es cualquier conjunto de *atributos que puede ser elegida* como una clave de una relación, la función de estas claves es la de proveer el mecanismo de direccionamiento a nivel de tupla en los sistemas relacionales.

La **clave primaria** es la clave candidata elegida como la clave de la relación, esta es escogida de entre las claves candidatas. Usualmente se prefiere la de menor tamaño.

Claves alternas o alternativas: el resto de claves candidatas no escogidas como clave primaria.

Una **clave externa** (ajena) es un conjunto de *atributos de una relación que es una clave en otra relación*, es decir, un atributo de una tabla (o agregación de ellos) puede ser clave primaria de otra tabla.

Los atributos de una clave ajena pueden o no formar parte de la clave primaria de la tabla a la que pertenecen. En una tabla no es obligatoria la existencia de claves ajenas.

El **esquema de una base de datos relacional** es un listado que muestra los nombres de las relaciones, los nombres de los atributos y las claves foráneas.

Restricciones de Integridad.

Una restricción es una regla que limita los valores que pueden estar presentes en la base de datos. El modelo de datos relacional de Codd incluye varias restricciones que se usan para verificar la validación de los datos en una base. Se considera las siguientes restricciones:

- a) **Integridad de la entidad.**
- b) **Integridad referencial.**
- c) **Dependencias funcionales.**

Las reglas de integridad de entidad y de referencia son, en definitiva, **restricciones que todas las relaciones deben cumplir obligatoriamente y por esta razón se las conoce como meta-reglas¹¹.**

Las restricciones de integridad proporcionan las bases lógicas para mantener la validación de los valores en la base de datos.

a) Regla de Integridad de la Entidad.

La clave de una relación identifica unívocamente cada fila y de aquí, cada instancia de la entidad. De este modo, si los usuarios quieren recuperar o manipular los datos almacenados en una fila específica, deben conocer el valor de la clave de esa fila. Eso significa que no se quiere que una entidad sea representada en la base de datos si no se tiene una identificación completa de los atributos que son claves en la entidad. De este modo no se permite que la clave, o una parte de la misma sea un valor nulo.

b) Regla de Integridad referencial.

La regla es: "La base de datos no puede contener claves foráneas no concordantes".

¹¹ Reglas que describen cómo deben ser utilizadas o modificadas las reglas que componen la base de conocimiento de un sistema de producción

Es decir, no puede existir una clave foránea cuyo valor (no nulo) no exista como clave primaria en alguna tupla de la relación referenciada.

Una base de datos en la cual todas las claves foráneas no nulas referencian valores reales de claves en otra relación cumple la integridad referencial.

c) Dependencias Funcionales.

Las dependencias funcionales proveen una manera para definir restricciones adicionales en un esquema relacional. La idea esencial es que el valor de la tupla en un atributo determina unívocamente el valor de la tupla en otro atributo.

“El valor de un atributo en una tupla determina el valor de otro atributo en la tupla”.

NORMALIZACIÓN.

La redundancia en los datos o repetición, no solo ocupa espacio, sino que puede conducir a perder la integridad de los datos (pérdida de la consistencia) en la base de datos.

Dicha inconsistencia podría ser en la información de las tuplas, ya sea en la actualización, borrado o inserción de las mismas.

La descomposición es el proceso de dividir relaciones en múltiples relaciones para eliminar dichas anomalías y mantener la integridad de los datos. Para hacer esto, se usan las formas normales o reglas para relaciones estructuradas.

La normalización es un proceso que consiste en comprobar que las tablas (también denominadas relaciones en terminología propia del modelo relacional de datos) definidas cumplen unas determinadas condiciones, es un método propio del modelo relacional y consiste en **descomponer las relaciones originales** en otras más pequeñas con el fin de **eliminar las anomalías** que se pueden dar en las relaciones iniciales. Con

esto se pretende garantizar la no existencia de redundancia y una cierta coherencia en la representación mediante un esquema relacional de las entidades y relaciones del modelo conceptual (diagrama Entidad – Relación).

A través de la normalización se pueden solucionar diversos errores en el diseño de la base de datos así como mejorarlo y facilitar el trabajo posterior del administrador de la base de datos y de los desarrolladores de aplicaciones.

Evidentemente las reglas de normalización imponen una serie de restricciones en lo relativo a la existencia de determinados esquemas relacionales. Según se avance en el cumplimiento de reglas y restricciones se alcanzará una mayor forma normal. Existen cinco formas normales (*figura 8*), hacia las cuales puede conducir el proceso de normalización de forma incremental más una forma normal independiente de las otras.

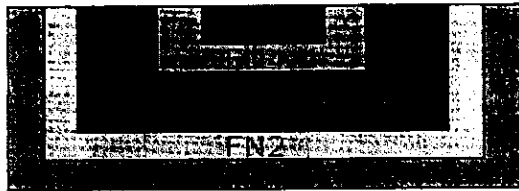


Figura 8. Formas Normales.

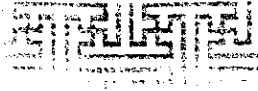
Un esquema relacional que satisface todas las restricciones impuestas por la tercera forma normal se considera de buena calidad aunque es mejor que satisfaga una interesante propiedad. La verificación de una forma normal implica el cumplimiento de todas las formas normales anteriores. La primera forma normal es de cumplimiento obligatorio para que exista siquiera un esquema relacional propiamente formado:

- **FN1. Atributos atómicos** Para cada atributo de la base, se requiere que sea atómico, es decir que no pueda descomponerse en sub-elementos. Por ejemplo, para personas se separa el nombre completo en Nombre(s) y

- **FN2. Separación de tablas** La segunda forma normal requiere además que todos los atributos en una tabla dependan completamente de la llave primaria de la tabla, es decir, que si alguno dependiera solo de una parte de la llave, este atributo deberá estar en otra tabla. En este sentido, se dice que un atributo depende de otro si, dado el valor del segundo, es posible saber el valor del primero. Si la clave está formada por un único atributo entonces ese esquema estará seguro en segunda forma normal.
- **FN3.** La tercera forma normal requiere además que cada atributo en la tabla dependa directamente de la llave primaria, es decir, que no pueda ser obtenido a partir de otros atributos dentro de la misma tabla (salvo los de la llave).
- **FNBC (Forma Normal de Boyce - Codd).** Se basa en el concepto de **determinante funcional**: uno o varios atributos de una tabla de los cuales dependen funcionalmente de forma completa algún otro atributo de la misma tabla. Una relación está en FNBC si FN1 y cada determinante funcional es una clave candidata de la tabla. Así se garantiza que se han elegido bien las claves al no existir dependencias funcionales entre atributos que no son clave.

Edgar Codd publicó dos artículos introduciendo el modelo de datos relacional y los lenguajes de manipulación de datos relacionales, el álgebra relacional y el cálculo relacional¹². Aunque el modelo de datos relacional era importante en sí mismo los lenguajes relacionales eran más significativos en cuanto a la revolución de las bases de datos relacionales.

¹² "A Relational Model of Data for Large Shared Data Banks" de la revista Communications de A.C.M Vol. 13 Nro. 6 de Junio de 1970.



CAPITULO III

LENGUAJE DE CONSULTA

ESTRUCTURADO (SQL)

Y

ARQUITECTURA CLIENTE

SERVIDOR.

- Álgebra relacional.
- Cálculo relacional.
- Lenguaje de Consulta Estructurado (SQL).
- Arquitectura Cliente / Servidor.
- Middleware.

Álgebra y Cálculo Relacional.

El **álgebra relacional** es un lenguaje procedimental para la manipulación de relaciones. Esto significa que el álgebra relacional usa una aproximación paso por paso para crear una relación que contenga los datos que responden a la consulta.

El **cálculo relacional** es no procedimental. En el cálculo relacional, una consulta se resuelve definiendo una relación en un simple paso.

Edgar Codd también mostró que el álgebra relacional y el cálculo relacional son lógicamente equivalentes, es decir, que cualquier consulta que podría ser formulada en el cálculo relacional podría ser formulada en el álgebra relacional y viceversa.

Álgebra relacional.

El **álgebra relacional** es también importante porque aporta mucho del vocabulario y muchos de los conceptos básicos de manipulación de datos relacionales que se encuentran comúnmente en los lenguajes de bases de datos comerciales. (*select, project, join y union*)

Las operaciones del álgebra relacional manipulan relaciones, esto es, que estas operaciones usan una o dos relaciones existentes para crear una nueva relación. Esta nueva relación puede entonces usarse como entrada para una nueva operación. Este poderoso concepto hace posible una variedad infinita de manipulaciones sobre los datos. Esto también hace considerablemente más fácil la solución de las consultas, debido a que se puede experimentar con soluciones parciales hasta encontrar una proposición con la que se trabajara.

El álgebra relacional consta de las siguientes nueve operaciones: union, intersección, diferencia, producto, selección, proyectar, renombrar, división y asignación.

Las cuatro primeras de estas operaciones se toman de la teoría de conjuntos de la matemática y que son considerablemente parecidas a las operaciones encontradas ahí

Esto es razonable, debido a que las relaciones son en sí mismas conjuntos, de esta manera se le pueden aplicar las operaciones de conjunto.

Las cuatro siguientes son operaciones nuevas que se aplican específicamente al modelo de datos relacional.

La última operación (asignación) es la operación estándar de los lenguajes de computación de dar un valor a un nombre. En este caso la asignación se usa para dar un nombre a una nueva relación que se crea de relaciones existentes. Se usará el signo " := " (es el nombre asignado a) para indicar la asignación de nombres a relaciones.

Consta de un conjunto de operaciones que actúan directamente sobre una ó dos relaciones para producir una nueva relación.

De acuerdo con Korth y Silberschatz, las operaciones fundamentales en el álgebra relacional son seleccionar, proyectar, producto cartesiano, renombrar, unión y diferencia de conjuntos. Adicional a las anteriores existe la intersección de conjuntos, el producto natural, la división y la asignación (esta última no se mencionará por su simpleza).

- ◆ **Seleccionar.** Operación unaria (σ) cuya función es seleccionar tuplas que cumplen con un predicado determinado. Su sintaxis es:

$$\sigma_{\text{predicado}} (\text{Relación})$$

- ◆ **Proyectar.** Operación unaria (π) cuya función es devolver tuplas univocas sobre ciertos atributos. Su sintaxis es:

$$\pi_{\text{columnas}} (\text{Relación})$$

- ◆ **Producto Cartesiano.** Operación binaria (χ). Establece las posibles combinaciones entre dos tipos de relaciones, aclarando además que en el formalismo una relación no es más que un subconjunto del producto cartesiano de un conjunto de dominios. Su sintaxis es:

$$\text{Relación } \chi \text{ Relación}$$

- **Renombrar.** Operación unaria (ρ). Es utilizada en aquellos casos donde puede existir ambigüedad en los nombres de relaciones durante la consulta. Su sintaxis es:

ρ nuevo_nombre(relación_original)

- **Unión.** Operación binaria (\cup). Entrega como resultado el conjunto de las tuplas contenidas en ambas relaciones que deben tener el mismo número de atributos y cada uno de estos con iguales dominios. Su sintaxis es:

Relación \cup Relación

- **Diferencia.** Operación binaria ($-$). Permite establecer las tuplas contenidas en una relación que no se encuentran en la otra, atendiendo a que ambas relaciones deban tener el mismo número de atributos, y estos a su vez con el mismo rango de dominios. La sintaxis de esta operación es la siguiente:

Relación - Relación

- **Intersección.** Operación binaria (\cap). Al igual que la anterior, exige que el grado sea igual y los dominios sean semejantes. Devuelve el conjunto de tuplas contenidas que coinciden con ambas relaciones. Su sintaxis es la siguiente:

Relación \cap Relación

- **Producto Natural.** Operación Binaria (γ). Tal como ha sido establecido por diversos autores, esta operación es la encargada de establecer las relaciones entre las entidades participantes de una consulta; ella sola realiza el producto cartesiano y determina las

coincidencias entre los atributos que permiten las coincidencias.
Su sintaxis es:

Relación1 y Relación2

que equivale a:

$\sigma_{\text{Relación1. atrib_rel}=\text{Relación2. atrib_rel}}(\text{Relación1} \times \text{Relación2})$

- **División.** Operación binaria (\div). Esta operación trabaja sobre una relación de grado 2 con otra de grado 1, quedando claro que esta relación de grado 1 debe tener un dominio semejante con alguno de los atributos de la relación que tiene grado 2. El resultado de esta operación es determinar las tuplas coincidentes sobre las instancias de las relaciones. Su sintaxis es la siguiente:

Relación \div Relación

Cálculo relacional.

El cálculo relacional usa un enfoque completamente diferente al álgebra relacional. No obstante, los dos lenguajes son lógicamente equivalentes.

Esto significa que cualquier consulta que pueda resolverse en un lenguaje puede resolverse en el otro. Se será más breve en el cálculo relacional, debido a que el lenguaje por sí mismo tiene menos construcciones.

El cálculo relacional es importante por **dos razones**:

- ◆ Esta basado en la **lógica formal** del cálculo de predicados, que es un método poderoso de determinación de la verdad de una instrucción a partir de la verdad de sus componentes. Consecuentemente el cálculo relacional tiene una fundamentación lógica tan firme como cualquier lenguaje de programación existente.

- ◆ Varios lenguajes relacionales comerciales están conceptualmente cercanos a él.

LENGUAJE DE CONSULTA SQL

Structured Query Language, Lenguaje de Consulta Estructurado.

En el modelo relacional, como se mencionó anteriormente los datos se representan como colecciones de tablas. Es el estándar industrial para acceder a las bases de datos relacionales.

Su origen está en el lenguaje SEQUEL¹³. Oracle fue el primer fabricante de sistemas de bases de datos en comercializar una implementación del Lenguaje de Consulta Estructurado en 1979. IBM lanzó el producto SQL/DS en 1981 y dos años más tarde el conocido DB2.

Se trata de un lenguaje fuertemente basado en el inglés, que puede ser utilizado en modo conversacional mediante un intérprete o bien formando parte de un programa desarrollado en un lenguaje de programación anfitrión como C¹⁴, Cobol¹⁵, etc. (Lenguaje de Consulta Estructurado embebido). La característica relacional más importante del Lenguaje de Consulta Estructurado es que permite acceder a los datos sin necesidad de especificar cómo se ha de realizar dicho acceso permitiendo así la "navegación automática" por los datos.

El lenguaje ha sido normalizado por varios organismos, si bien los muchos aspectos que tradicionalmente han quedado indefinidos en tales normalizaciones y que por lo tanto en la práctica quedan como responsabilidad de los fabricantes han ocasionado variaciones de diversa importancia entre las diferentes implementaciones comerciales.

El lenguaje SQL, siglas que provienen de *Structured Query Language* (Lenguaje de Consulta Estructurado) es un lenguaje de bases de datos que pretende ser un lenguaje común para las bases de datos relacionales.

¹³ (*Structured English QUERy Language*) desarrollado en IBM en los años 1974-75.

¹⁴ Lenguaje de programación estructurado, de aplicación general.

¹⁵ *COmmon Business Oriented Language*. Es un lenguaje de programación que desarrolló CODASYL y, de hecho, es norma en el proceso de datos comerciales.

El Lenguaje de Consulta Estructurado consta de dos partes: el DDL o *Data Definition Language* (Lenguaje de Definición de Datos) y el DML o *Data Manipulation Language* (Lenguaje de Manipulación de Datos)

Comandos y Descripción.

Creación de tablas.

(CREATE TABLE) (DDL)

```
CREATE TABLE CensoMunicipal
(
  codprov VARCHAR(2) PRIMARY KEY,
  nomprov VARCHAR(20) NOT NULL
)
```

Borrado de tablas

Borrado de tablas (DROP TABLE) (DDL)

```
DROP TABLE nombreTabla
```

Modificación de la estructura de las tablas

(ALTER TABLE) (DDL)

```
ALTER TABLE CensoMunicipal ADD (horas NUMBER(4,2))
```

También pueden eliminarse atributos ya definidos

```
ALTER TABLE nombreTabla DROP nombreAtributo
```

Introducción o inserción de tuplas

Modificación de la estructura de las tablas (ALTER TABLE) (DDL)

```

INSERT INTO alumno VALUES (12, 21, 'GÓTERA OTILIO', 'BENITO', 19, 3, 28)
INSERT INTO alumno VALUES (13, 21, 'GÓTERA OTILIO', 'BENITO', 19, 3, 28)
INSERT INTO alumno VALUES (14, 21, 'GÓTERA OTILIO', 'BENITO', 19, 3, 28)
INSERT INTO alumno VALUES (15, 21, 'GÓTERA OTILIO', 'BENITO', 19, 3, 28)

```

Para que la inserción de una tupla sea válida deben satisfacerse las restricciones de clave, integridad y de otro tipo que pudieran existir. Es importante que los valores de los atributos se especifiquen en el mismo orden en que fueron declarados los atributos. Si no se recuerda el orden de los atributos entonces se puede especificar en cualquier orden el nombre de cada atributo dentro de la sentencia INSERT de la siguiente manera:

```

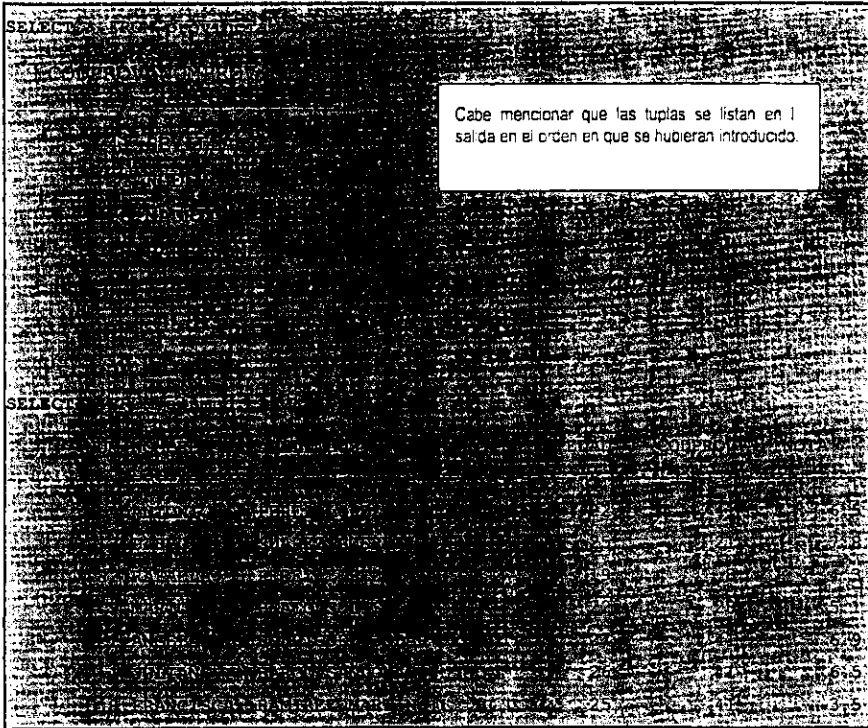
INSERT INTO alumno (num, codprov, apellidos, nombre, nota, edad)
VALUES (12, 21, 'GÓTERA OTILIO', 'BENITO', 19, 3, 28)

```

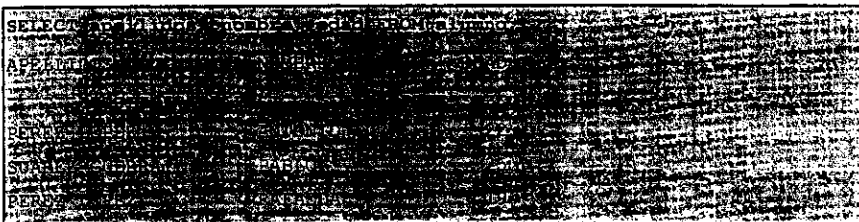
Consulta

(SELECT) (DML)

1. Listar todo el contenido de una tabla:



2. Selección de atributos concretos:



3. Ordenación de la salida:

The screenshot shows a SQL query: `SELECT * FROM PERSONA ORDER BY apellido`. Below the query, there is a text box with the following text: "Puede ordenarse tanto por valores alfabéticos como numéricos. Puede ordenarse de menor a mayor (orden ascendente, o ASC) o de mayor a menor (orden descendente, o DESC). Por defecto el orden considerado es de menor a mayor. La ordenación tiene asociado un cierto coste computacional por lo que solo debe realizarse cuando sea necesario." Below the text box, the output of the query is visible, showing columns for 'Apellido', 'Nombre', 'Sexo', 'FechaNacimiento', and 'FechaAlta', with rows for 'Alfonso', 'Pedro', 'Beatriz', and 'Sofía'.

4. Ordenación múltiple:

The screenshot shows a SQL query: `SELECT * FROM PERSONA ORDER BY apellido, nombre`. The output shows the results ordered first by 'Apellido' and then by 'Nombre' within each group.

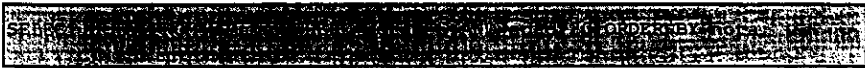
5. Condición simple. Consulta condicional (WHERE)

The screenshot shows a SQL query: `SELECT * FROM PERSONA WHERE apellido = 'Alfonso'`. The output shows only the rows where the 'Apellido' is 'Alfonso'.

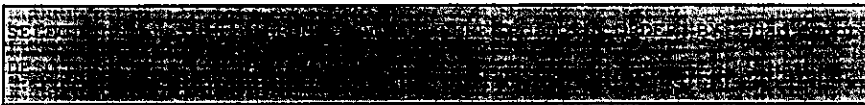
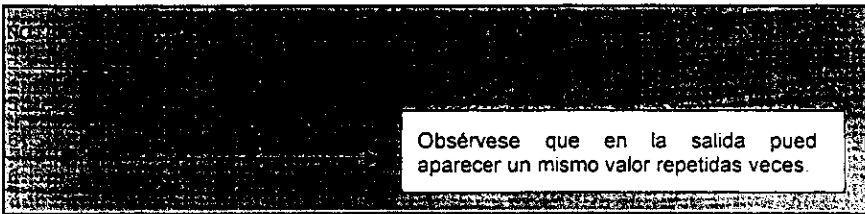
6. Condición compuesta



7. Combinación de condición y ordenación Ordenación de la salida (ORDER BY)



(Nombre y apellidos de los alumnos aprobados (nota>=5) y con edad mayor de 20 años, ordenados según la nota).



(Nombre y apellidos de los alumnos con edades mayores de 20 años, ordenados por edad y nota).

8. Evitar repetición en la salida

Evitar repetición en la salida (DISTINCT)



La salida de una consulta de este tipo suele (dependiendo de la implementación concreta en cada caso) estar ordenada por defecto.

9. Función media aritmética

Funciones de agregación (AVG, MAX, MIN, SUM, COUNT)

Las funciones de agregación toman como entrada una colección de valores o tuplas de salida producidas por una consulta y producen un único valor como salida.

(AVG)



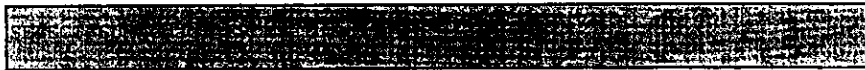
10. Función de cuenta (COUNT)



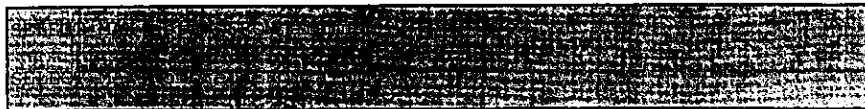
11. Cardinalidad de una tabla (número de filas o tuplas)



12. Función de suma (SUM)



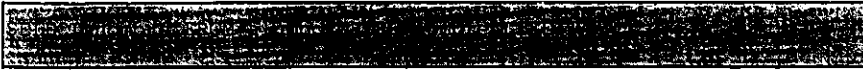
13. Función de máximo (MAX)



14. Función de mínimo (MIN)



15. Nombres de alumnos que empiezan por 'A'



Esta comparación es sensible a minúsculas y mayúsculas. El carácter '*' equivale al comodín asterisco '*' del DOS¹⁸: se ignora un número variable de caracteres. El carácter '_' equivale al comodín '?' del DOS: se ignora un carácter. Para revisar la no presencia de patrones en cadenas se puede utilizar NOT LIKE.

Combinación de consultas

Operaciones sobre conjuntos

Operación	Descripción
UNION	Unión de conjuntos (usualmente resultados de salida).
INTERSECT	Intersección de conjuntos.
EXCEPT MINUS	Diferencia de conjuntos.

Los conjuntos que participen en este tipo de operaciones han de ser compatibles, esto es, deben tener el mismo conjunto de atributos o resultados de salida. Estas tres operaciones eliminan automáticamente los duplicados (se supone que un elemento solo puede estar una vez en un conjunto).

¹⁸ *Disk Operating System*. Sistema Operativo basado en Discos. Sistema operativo monousuario y monotarea (actualmente) para ordenadores personales IBM y compatibles

Permisos sobre tablas

Por otra parte, los usuarios son directos responsables y propietarios de las tablas que eventualmente creen y pueden a su vez otorgar privilegios de varios tipos sobre esas tablas a otros usuarios según su voluntad. Por defecto un usuario solo tiene acceso a las tablas que él mismo haya creado y no dispone de ningún permiso sobre las tablas de los otros usuarios. Se puede referenciar directamente una tabla de otro usuario de la forma: nombre_usuario.nombre_tabla. Los permisos asociados a las tablas y que pueden ser concedidos y retirados por el creador son los siguientes:

SELECT	Consulta de tuplas
INSERT	Introducción de tuplas
UPDATE	Modificación de tuplas
DELETE	Borrado de tuplas
ALTER	Modificación de la estructura
GRANT	Todos los permisos

Vistas

Las vistas constituyen un poderoso mecanismo del modelo relacional de datos en lo referente a la interacción entre el usuario o las aplicaciones y el esquema relacional o conjunto físico de tablas. Las vistas posibilitan el nombrar de diferentes formas a los mismos datos, constituyendo una especie de tablas virtuales o ventanas a través de las cuales se puede ver los datos realmente almacenados en las tablas. Las ventajas fundamentales que conlleva la utilización de vistas son las siguientes:

Privacidad y seguridad: no todos los usuarios pueden ver o conocer la totalidad del esquema relacional. Pueden existir datos ocultos a unos usuarios y conocidos para otros según limitaciones referentes a la información relevante para los usuarios.

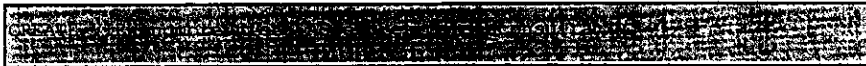
Permiten limitar las posibles operaciones (consultas) que se puedan realizar sobre los datos.

Simplificación del interfaz con el usuario

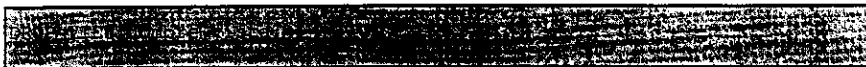
Permiten denominar a una operación compleja con un nombre simple y expresivo bajo el cual pueda ser invocada más fácilmente.

Favorecer la independencia lógica de los datos. Se puede variar la estructura de la base de datos sin necesidad de efectuar cambios profundos en los programas, sino tan solo en las vistas.

Las vistas se definen mediante la sentencia `CREATE VIEW` de la siguiente manera:



Las vistas se eliminan (tan solo su definición, no las tablas sobre las que se construye) mediante la sentencia `DROP VIEW`:



Una vista se especifica a través de una expresión de consulta (una sentencia `SELECT`) que la calcula y que puede realizarse sobre una o más tablas. Sobre un conjunto de tablas relacionales se puede trabajar con un número cualquiera de vistas.

CLIENTE / SERVIDOR.

Con la proliferación de ordenadores personales de bajo coste en el mercado, los recursos de sistemas de información existentes en cualquier organización se pueden distribuir entre ordenadores de diferentes tipos: ordenadores personales de gama baja, media y alta, estaciones de trabajo, miniordenadores¹⁷ o incluso grandes ordenadores.

El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos estos recursos de máquina de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe¹⁸ se traspasa a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

A continuación se describen con más detalle tanto a la parte del Cliente como a la del Servidor.

Cliente

El programa **cliente** (*figura 9*), cumple dos funciones distintas: 1. Gestiona la comunicación con el servidor, solicita un servicio y recibe los datos enviados por aquél; 2. Maneja la interfaz con el usuario: presenta datos en el formato adecuado y da herramientas y comandos necesarios para el usuario.

Los clientes realizan generalmente funciones como:

¹⁷ Ordenador de gran potencia que se sitúa entre la categoría de las estaciones de trabajo y de los *mainframes*. En general, suelen estar basados en chips y sistemas operativos no abiertos, aunque UNIX está penetrando con fuerza en este mercado.

¹⁸ **Mainframe**. Ordenador central con gran capacidad de proceso y de almacenamiento. Sistema de proceso orientado a transacciones de base de datos, capaz de dar servicio a cientos o miles de usuarios simultáneamente.

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

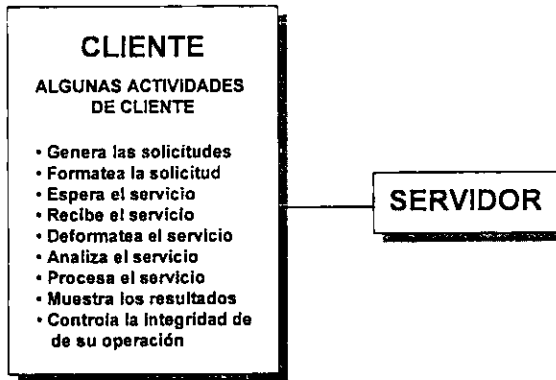


Figura9.
El Cliente genera requerimientos de servicio y espera el requerimiento.

Servidor.

Sólo tiene que encargarse de transmitir la información de forma eficiente (Figura10). No tiene que atender al usuario. Puede atender a varios clientes al mismo tiempo, realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

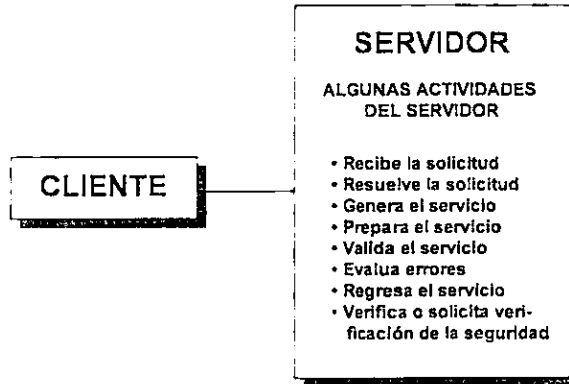


Figura 10.

El **servidor** recibe requerimientos de servicio, ejecuta el proceso (llama al programa o rutina aplicativa destino) y elabora una respuesta al requerimiento.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo (figura 11).

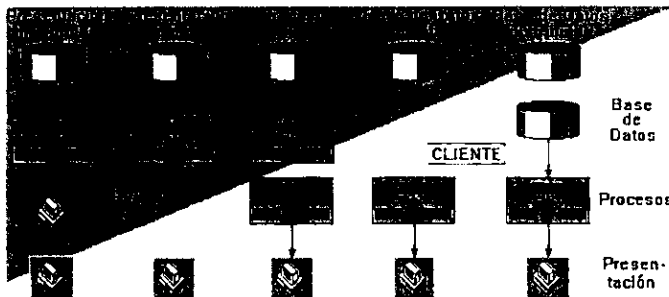


Figura 11.

Base de Datos. Proceso y Presentación Arquitectura Cliente/Servidor

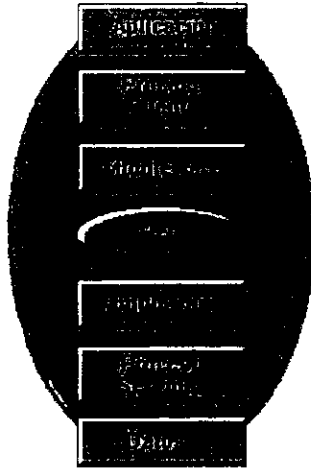


Figura12.

Procesos Arquitectura Cliente/Servidor.

La arquitectura cliente/servidor no exige requisitos especialmente complejos a los Sistemas de Gestión de Bases de Datos ya que, aunque estén involucrados varios ordenadores, la base de datos en sí está normalmente centralizada en un ordenador y su mantenimiento es igual de sencillo que en una arquitectura centralizada clásica. Para esta arquitectura es importante que el Sistema de Gestión de Bases de Datos soporte sistemas de comunicación normalizados ya que tendrá que recibir peticiones de diversos clientes operando con máquinas y protocolos distintos (figura 12).

MIDDLEWARE.

Equipo lógico intermedio que permite la comunicación entre dos módulos de *software*.

El middleware proveerá los niveles de seguridad que se necesitan para mantener unos altos estándares de integridad de la información y una completa seguridad de que la información está siendo utilizada por la persona adecuada en la tarea adecuada. También garantizará que los planes de contingencia que se tengan sean viables y que se cuente con la infraestructura necesaria para colocarlos en práctica oportunamente.

Dentro de las principales características que debe cumplir un middleware que apoye a la administración de la empresa se deben garantizar las siguientes:

- ◆ Balancear las cargas de trabajo entre los elementos de computación disponibles.
- ◆ Manejo de mensajes, que le permite entrar en el modo conversacional de un esquema Cliente/Servidor y en general de cualquier forma de paso de mensajes entre procesos.
- ◆ Administración Global, como una sola unidad computacional lógica.
- ◆ Manejo de la consistencia entre los diferentes manejadores de bases de datos principalmente en los procesos de OLTP¹⁹.
- ◆ Administración de la alta disponibilidad de la solución.

El "middleware" es un **componente de software que está entre el sistema operativo y la aplicación**, permitiendo al programador aislarse de las particularidades de sistema operativo y del ambiente de redes que se está utilizando. Las complejidades del mundo de las comunicaciones y las peculiaridades de los diversos sistemas operativos son algo que se encarga de manejar la gente que desarrolla el "middleware". El

¹⁹ **OLTP** *On Line Transaction Processing* Proceso transaccional en línea. Método de proceso continuo de transacciones.

programador lo que hace es comprar este software de "middleware" e incorporarlo a la aplicación que está desarrollando. De igual manera, cada vez que sale al mercado una nueva versión de un sistema operativo la empresa que produce el software de "middleware", se encarga de asegurar que este "middleware" sea actualizado para que el desarrollador de aplicaciones lo pueda incluir en su aplicación sin tener que saber ni siquiera qué cambió en la nueva versión del sistema operativo. Además la aplicación es más robusta y eficiente si utiliza un "middleware" que se hace a nivel mundial que si trata de desarrollar esta experticia a nivel local. De esta manera, el escribir una aplicación utilizando un buen "middleware", permite al desarrollador hacer que su lógica del negocio no sólo sea Cliente/Servidor, sino que además la hace independiente de los sistemas operativos y los diversos protocolos de comunicaciones, aumentando considerablemente el espectro de posibles compradores.

Los sistemas relacionales de hoy aún están evolucionando y, en algunos aspectos significativos, cambiando su naturaleza subyacente para permitir a los usuarios plantear problemas más complejos. Uno de los cambios más importantes está ocurriendo en el área de las bases de datos orientadas a objetos.



CAPITULO IV

CASO PRACTICO

"BASE DE DATOS PARA LA ADMINISTRACION Y CONTROL EN EL ACCESO A LOS SISTEMAS JURÍDICOS."

- ◆ Planteamiento del problema.
- ◆ Solución al Proyecto.
- ◆ Sistema Prototipo.

BASE DE DATOS PARA LA ADMINISTRACIÓN Y CONTROL EN EL ACCESO EN LOS SISTEMAS JURÍDICOS.

Planteamiento del Problema.

En el **Área de Seguridad de Datos** existe la preocupación de asegurar la información sensible de todos los **Usuarios** existentes en los diferentes **Sistemas** de la **Institución**, ya que por las dimensiones del personal con que cuenta y la diversidad de **Sistemas**, es indispensable que se cuente con un Sistema que permita tener pleno control del personal que se encuentra dado de Alta en cada Sistema y a qué recursos tiene acceso.

Es indispensable que la base de datos de éste sistema registre a los empleados de la Institución actualizando constantemente **Área, Puesto, Funciones, Bajas, Perfiles** y demás cambios posibles, garantizando así la prevención en el *mal uso o préstamo de claves de personal*, otorgando un servicio con calidad en las **Altas, Bajas, Cambio de Transacciones, Cambio de Perfiles, y Cambio de Roles** a todos los Usuarios de los sistemas.

Actualmente se tiene la administración de **4 sistemas de Derecho Penal Institucional** con **1000 Usuarios**, los cuales se incrementan día con día, debiendo ser identificados, manejados y almacenados, de tal modo que siempre estén actualizados y disponibles para una red de computadoras en todo momento.

La implantación de éste sistema pretende controlar tanto a los **Usuarios** como a los **Sistemas** que se Administran, incrementando con ello la confiabilidad en los procesos de la Institución.

Requerimientos mínimos.

De acuerdo a las necesidades de la Institución, los datos mínimos requeridos de cada Usuario para su registro deben ser los siguientes:

- Número de empleado.
- RFC.
- Nombre.
- Centro Regional.
- División.
- Dirección.
- Subdirección.
- Área.
- Puesto.
- Plaza.
- Comentario.

Los sistemas que debe de administrar el **Sistema de Seguridad** y de los cuales debe llevar el control de accesos de Usuarios, debido al alto grado de criticidad de la información que manejan son los siguientes:

- Sistema de Averiguaciones Previas .
- Sistema de Control de Causas Penales.
- Sistema de Control de Amparos.
- Sistema de Control de Operaciones Judiciales.

Este sistema debe controlar tanto a los Usuarios como a los sistemas administrados y debe de contar con módulos que permitan el control de las **Altas, Bajas, Consultas y Modificaciones** de cada Usuario.

Ambiente de trabajo.

Se cuenta con computadoras personales Pentium, una Red, Windows 95 y Windows NT, con 32 Mb en Memoria Ram.

Solución al Proyecto.

La solución propuesta para el problema que anteriormente se ha planteado consistirá en el desarrollo de un sistema que cumpla con la funcionalidad solicitada y el adecuado diseño de la base de datos que permita registrar toda la información necesaria, permitiendo proporcionar un sistema confiable y oportuno.

Para que el sistema desarrollado cumpla con su objetivo haré uso de las siguientes herramientas:

- ☛ SQL Server versión 6.5 como manejador de base de datos, por medio del cuál definiré, crearé y poblaré las relaciones necesarias.
- ☛ Visual Basic Enterprise Edition Versión 5.0 como lenguaje de desarrollo.
- ☛ El sistema trabajará bajo Windows 95 en las estaciones de trabajo.
- ☛ Y en el servidor estará corriendo Windows NT versión 3.51

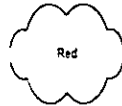
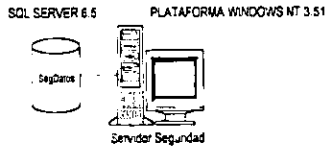
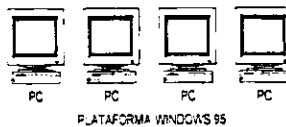


Diagrama de Conexión de las herramientas para el proyecto.

AREA DE SEGURIDAD DE DATOS



Definición de la base de datos.

De acuerdo a las necesidades que se han planteado será necesario contar con una base de datos que se llame **SEGDATOS**, en la cuál se registre la información. Esta base de datos se conformará de las siguientes relaciones:

T001_CDIRECCION.

Es un catálogo en el cual se guarda la Clave y el Nombre de cada Dirección existente en la Institución.

Atributo	Tipo
Cd_Dirección	Char 6
Nb_Direccion	Barchar 35

T002_CDIVISION.

Este es un catálogo en el cual se guarda la Clave y Nombre de cada División existente en la Institución.

Atributo	Tipo
Cd_Division	Char 6
Nb_Division	Barchar 35

T003_CSUBDIRECCION.

En esta relación se guarda la Clave y Nombre de cada Sub-Dirección existente en la Institución, el cual es utilizado como un catálogo para el sistema.

Atributo	Tipo
Cd_SubDireccion	Char 6
Nb_SubDireccion	Barchar 35

T004_CPUESTO.

En este se guarda la Clave y Nombre de cada **Puesto** existente en la Institución.

Atributo	Tipo
Cd_Puesto	Char 6
Nb_Puesto	Barchar 35

T005_CAREA.

Para guardar la Clave y el Nombre del **Área** en la Base de Datos se empleó esta relación.

Atributo	Tipo
Cd_Area	Char 6
Nb_Area	Barchar 35

T006_CENT_REGIONAL.

Este es un catálogo en el cual se guarda la Clave y Nombre de cada **Centro Regional** que hay en la Institución.

Atributo	Tipo
Cd_Cent_Regional	Char 6
Nb_Cent_Regional	Barchar 35

T007_DUSUARIO.

Esta es la tabla principal y en ella se guardan los diferentes datos del usuario como son:

Atributo	Tipo
Cd_RegUsuario	Char 8
Nb_Nombre	Barchar 45
Cd_RFC	Char 15
Nb_Plaza	Barchar 35
Cd_Area	Char 6
Cd_Puesto	Char 6
Cd_Cent_Regional	Char 6
Cd_Division	Char 6
Cd_Direccion	Char 6
Cd_SubDireccion	Char 6
Fec_Alta	Datetime
Fec_Baja	Datetime
Tx_Comentario	Text
Tp_Activo	Char 2

T008_D_Ave_Pre.

Este es la Tabla que representa los datos de los usuarios que tiene acceso al sistema de **Averiguaciones Previas**, en ella se guardan los siguientes atributos:

Atributo	Tipo
Cd_Reg_Usuario	Char 6
Cd_Usuario	Char 8
Nb_password	Char 8
Fec_Alta	Datetime
Fec_Baja	Datetime
Cd_Perfil	Char 6
Tx_Comentario	Text
Tp_Activo	Char2
Cd_Admon	Char 8

T09_D_Cau_Pen.

La siguiente Tabla registra los datos de los usuarios que manejarán el Sistema de las Causas Penales

Atributo	Tipo
Cd_Reg_Usuario	Char 6
Cd_Usuario	Char 8
Nb_password	Char 8
Fec_Alta	Datetime
Fec_Baja	Datetime
Cd_Perfil	Char 6
Tx_Comentario	Text
Tp_Activo	Char 2
Cd_Admon	Char 8

T010_D_Con_Pro.

Para guardar los datos de los usuarios que manejarán el sistema de Control de Procesos se emplea la siguiente tabla:

Atributo	Tipo
Cd_Reg_Usuario	Char 6
Cd_Usuario	Char 8
Nb_password	Char 8
Fec_Alta	Datetime
Fec_Baja	Datetime
Cd_Perfil	Char 6
Tx_Comentario	Text
Tp_Activo	Char 2
Cd_Admon	Char 8

T011_D_OpeJud.

Los datos de los usuarios que trabajen en el sistema de **Operaciones Judiciales**, se encontrarán registrados en la siguiente tabla:

Atributo	Tipo
Cd_Reg_Usuario	Char 6
Cd_Usuario	Char 8
Nb_password	Char 8
Fec_Alta	Datetime
Fec_Baja	Datetime
Cd_Perfil	Char 6
Tx_Comentario	Text
Tp_Activo	Char 2
Cd_Admon	Char 8

T012_DPERFILES.

Este es un catálogo en el cual se guarda la clave y nombre de cada Perfil de usuario existente en la Institución.

Atributo	Tipo
Cd_Perfil	Char 6
Nb_Perfil	Barchar 25

T013_SGRUPO.

Para que los usuarios puedan acceder a los diferentes sistemas, deben pertenecer a un grupo en específico, en esta tabla se guardarán los **Grupos** existentes en la relación para poder emplearla como un catálogo.

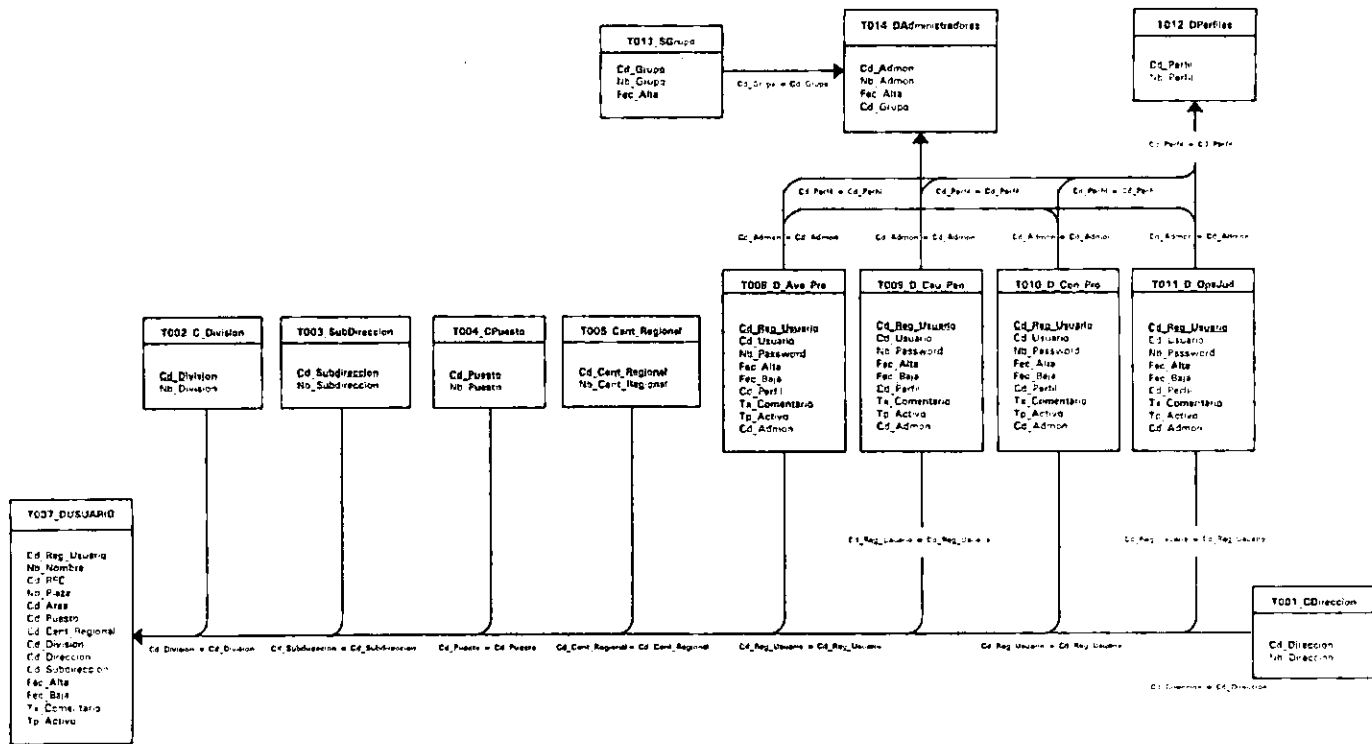
Atributo	Tipo
Cd_Grupo	Char 6
Nb_Grupo	Barchar 25
Fec_Alta	Datetime

T014_ADMINISTRAD.

Esta es una Tabla en la cual se guardan los datos de los **Administradores** que se han dado de alta para poder operar éste sistema.

Atributo	Tipo
Cd_Admon	Char 8
Nb_Admon	Barchar 35
Fec_Alta	Datetime
Cd_Grupo	Char 6

Para poder tener una mejor representación de la base de datos del proyecto, a continuación se plasma la forma en que deberán estar relacionadas las tablas:



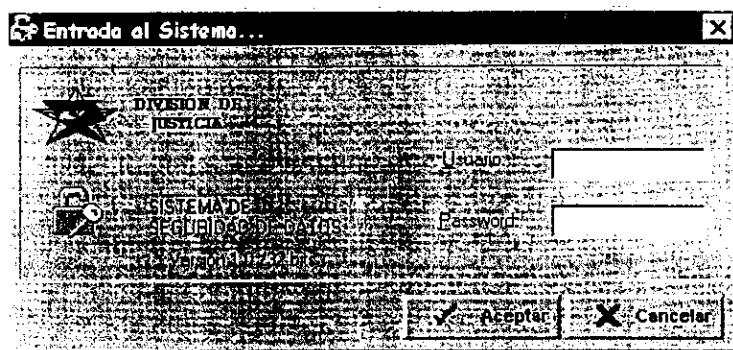
Tablas y Relaciones

Sistema Prototipo.

El sistema que se desarrollará estará conformado con las siguientes pantallas y funciones:

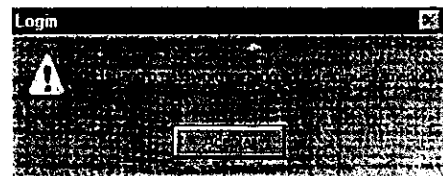
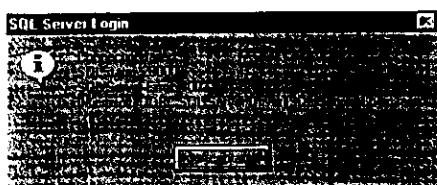
Pantalla Inicial.

Esta es la entrada al sistema y es mediante esta pantalla como los diversos administradores podrán tener acceso al Sistema de Seguridad de Datos. La clave del Usuario y el Password puede escribirse indistintamente (mayúsculas o minúsculas). Cabe mencionar que existe una tabla dentro de la base de datos del sistema, en la que se guardan todos y cada uno de los datos de los administradores que tienen acceso a este sistema, pero sin guardar el password de acceso.



Pantalla de Entrada al Sistema para los administradores

En caso de existir algún error al momento de teclear tanto la clave de Usuario como el Password, los mensajes que el sistema desplegaría son los siguientes:



Pantallas en caso de error en el acceso.

Pantalla Principal del sistema.

En esta pantalla se cuenta con los menús que permitirán al usuario administrar y controlar los usuarios permitidos en cada uno de los cuatro sistemas de Seguridad de Datos.



Pantalla Principal del sistema.

A continuación se detallan cada una de las Opciones de la barra de menús:

1) Menú Usuarios/Sistemas.

En este menú se tienen dos opciones, la del Mantenimiento de Usuarios y la de Usuarios de Sistema. a continuación se detallan cada una de las opciones:

Opción Mantenimiento de Usuarios.



Pantalla del primer menú Usuarios/Sistemas.

Al acceder a la primer opción que es **Mantenimiento de Usuarios** nos despliega una pantalla en la que contiene dos Pestañas, la primera de ellas referente a los Datos Generales de cada usuario, como a continuación se muestra:

Pantalla Inicial del menú de Mantenimiento de Usuarios

Dentro de esta opción, el administrador podrá realizar las siguientes acciones: **Alta, Baja, Modificación y Consulta**. Estas acciones se efectúan sobre la base de datos.



Para realizar la Alta de un Usuario se requieren capturar los siguientes datos:

- 1) Número de registro.
- 2) R.F.C.
- 3) Nombre
- 4) Centro Regional.
- 5) División.
- 6) Dirección.

- 7) Sub - Dirección.
- 8) Área.
- 9) Puesto.
- 10) Plaza.
- 11) Comentarios.

De los puntos 4 al 9 el sistema cuenta con catálogos creados en la base de datos con la finalidad de facilitar al Administrador la captura de datos del nuevo Usuario.



Para dar de baja a un Usuario basta con teclear el número de registro que se desea dar de baja y presionar el botón Aceptar inmediatamente después, el sistema confirmará la baja del usuario, en caso de seleccionar SI, el sistema dará de baja definitivamente al usuario, en caso de seleccionar NO, el usuario permanecerá activo en el sistema.



Si se desea efectuar algún cambio en los datos del usuario, el sistema habilitará la información que tiene dada de alta hasta ese momento y el administrador que vaya a realizar la operación de modificación, podrá cambiar la información que requiera.



En esta opción el administrador podrá consultar (sin modificar) la información que tengan uno o más usuarios.

La siguiente pantalla es un ejemplo de los datos de un usuario al momento en que un administrador realiza una Consulta.

Mantenimiento a Usuarios CONSULTA

DIVISION DE JUSTICIA

Datos Generales | **Sistemas**

Número de Registro: 01799830

C.I.R.: LALR740415

Nombre: RENE RUBEN LAZCANO LOPEZ

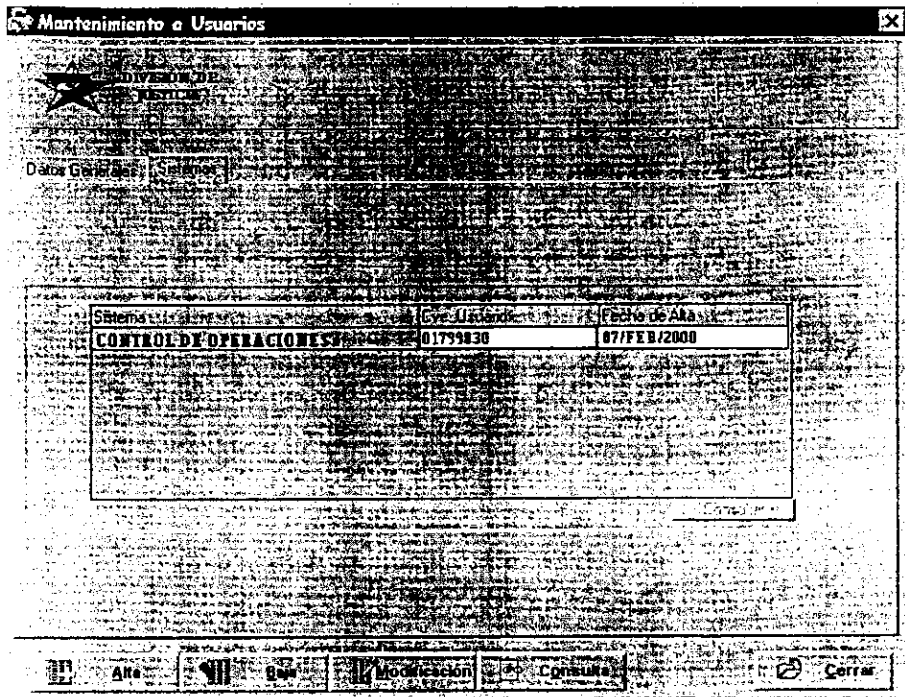
Centro Registral: 01	JALISCO	Area: 102479	PROCESOS
Código: 000117	CORD. GRAL. COMPUTO	Plaza: 2102	ANALISTA
Dirección: 001024	DIRECCION GENERAL		
Sub-Dirección: 010247	SISTEMAS Y TELECOM.		

Municipio: PUERTO VALLARTA

Comentarios:

Aceptar **Cancelar**

Pantalla de Consulta. Pestaña Datos Generales.



Pantalla de Consulta, Pestaña Sistemas.

Esta pestaña muestra el/los Sistema(s) en donde se encuentra dado de Alta el Usuario consultado o bien el sistema en el que se encuentra activo. Los tres datos que muestra del Sistema son:

- Nombre del Sistema.
- Clave del Usuario asignada.
- Fecha en que fue dado de alta en el sistema correspondiente.

El administrador podrá ir a consultar el detalle de la información del Sistema elegido sin poder modificar la información, para ello bastará con dar doble click en algún renglón de la cuadrícula mostrada o bien colocarse en el renglón deseado y presionar el botón consultar que presenta la siguiente pantalla.

SISTEMA DE CONTROL DE OPERACIONES

DIVISION DE
TELCEL

01799830

RENE RUBEN LAZCANO LOPEZ **LALR740415**

JALISCO **PROCESOS**

CORD. GRAL. COMPUTO **ANALISTA**

DIRECCION GENERAL

SISTEMAS Y TELECOM.

PUERTO VALLARTA

01799830 ***** 07/FEB/2000

ASESOR

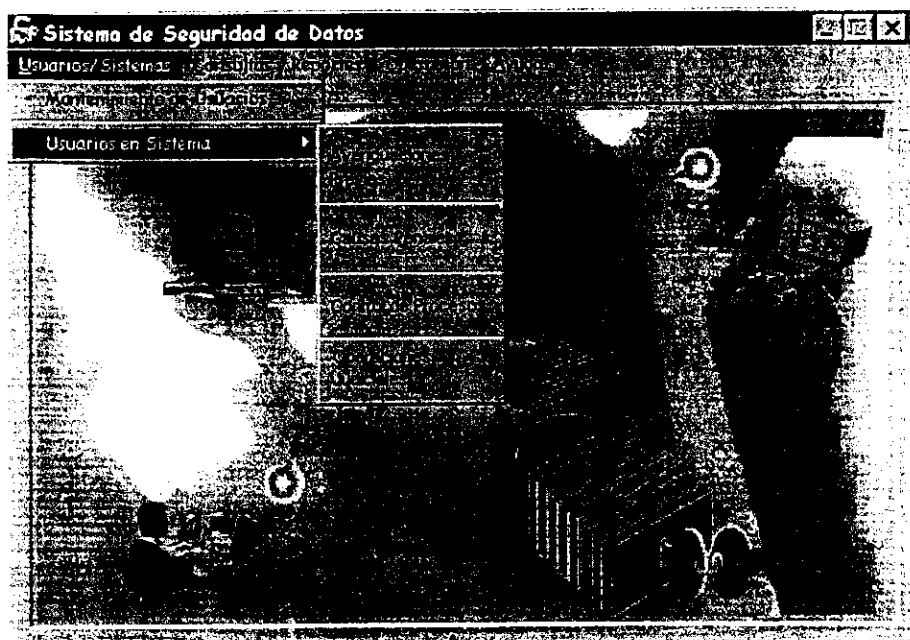
X Cancelar

Pantalla de Consulta del Sistema de Control de Operaciones

Es importante hacer notar que dependiendo del sistema que se esté consultando, en la parte inferior de la pantalla se mostrarán los datos particulares del usuario en el sistema.

Opción Usuarios en Sistema.

Esta opción permitirá al administrador del Sistema dar de Alta, Baja, Modificar o Consultar a un Usuario dentro de uno o más sistemas. Los sistemas que se encuentran implementados dentro de éste Sistema son los siguientes:



Pantalla de Usuarios/Sistemas.
En ella están cuatro de las opciones Jurídicas más importantes de la Institución

Al acceder a cada uno de éstos Sistemas se desplegará una pantalla en la que deberá capturarse la información correspondiente para la debida administración de los usuarios que tendrán acceso, a continuación se mostrará una pantalla de ejemplo correspondiente al módulo de Averiguaciones Previas:

La siguiente pantalla es un ejemplo al acceder a los datos del usuario del Sistema de Averiguaciones Previas.

The screenshot shows a window titled "Sistema de Averiguaciones Previas". Inside the window, there is a form with the following fields:

- A name field with the text "DIVISION DE INVESTIGACIONES" above it.
- A date field with a dropdown arrow.
- A large text area for additional information.
- At the bottom, there are three buttons: "Aceptar", "Cancelar", and "Salir".

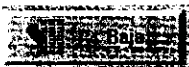
Datos del Usuario dentro del Sistema de Averiguaciones Previas



Con esta acción el administrador podrá dar de Alta a un Usuario en el Sistema solicitado. Dados los diferentes sistemas y el requerimiento del sistema, algunos de los campos a capturar por cada uno de los sistemas, serán obligatorios y otros opcionales. Permitiéndole así al administrador tener mayor consistencia en la información que se esta almacenando en la base de datos.

Cabe mencionar que un Usuario no podrá ser dado de Alta más de una vez. Los datos constantes en cada uno de los cuatro sistemas son:

- ◆ **Clave del Usuario:** Es la clave que el administrador asignará al Usuario para acceder al sistema solicitado.
- ◆ **Password:** Clave de acceso que el administrador asignará al usuario. Cabe mencionar que esta clave, no aparece en claro al momento de capturarla sino que aparece con astenscos.
- ◆ **Fecha de Alta:** Es la fecha del día actual en la que se efectuó el Alta del Usuario, en éste Sistema, la Fecha se toma del Servidor y no se le permite al administrador la modificación de ésta.
- ◆ **Perfil:** Es el perfil que el administrador asigna al Usuario dadas las características solicitadas por el mismo y dados los niveles de autorización a otorgar para cada una de las aplicaciones. Los perfiles para cada uno de los sistemas son distintos y se muestran en una lista de selección por cada Sistema. En ocasiones no se emplea una lista dado que el perfil se forma de cajas de selección múltiple.



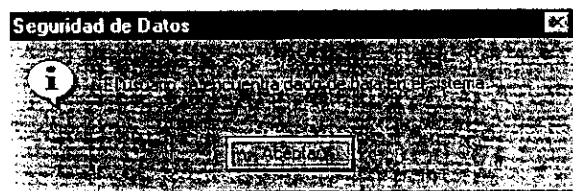
Un usuario podrá ser dado de baja de un sistema en específico dadas diversas circunstancias como renuncia, despido, etc.

Una vez que se presiona éste botón el sistema confirmará su eliminación, por lo que en caso de elegir "Sí", el sistema marca la baja con la leyenda: "**Usuario dado de Baja**" y adicional a esto, el sistema muestra la fecha en que fue realizada la baja del usuario.

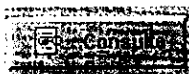


Una vez que el administrador dio de Alta a un Usuario, se podrán efectuar cambios en la información que inicialmente fue asignada, éstos cambios pueden darse debido a cambio de área puesto, perfil, etc.

Es importante mencionar que la información se sobre escribe a la ya existente, ya que después de haber efectuado alguna modificación, ésta inmediatamente se verá reflejada en la pantalla. Cuando el administrador intenta modificar la información de un Usuario que previamente fue dado de baja el sistema despliega el siguiente mensaje:



Esto significa que un usuario que se encuentre dado de baja, no se le podrá modificar su información.



Como su nombre lo indica, con ésta acción, el administrador podrá consultar a los Usuarios en determinado Sistema. Esta opción permite consultar a Usuarios tanto activos como inactivos; es decir, Usuarios dados de Alta y Usuarios dados de baja.

Toda consulta que se realice, impide al administrador modificar la información que muestra el Sistema.



Con esta función el administrador abandona la pantalla del Sistema en el que se encuentre, regresando siempre a la pantalla principal.

2) Menú Consultas y Reportes.

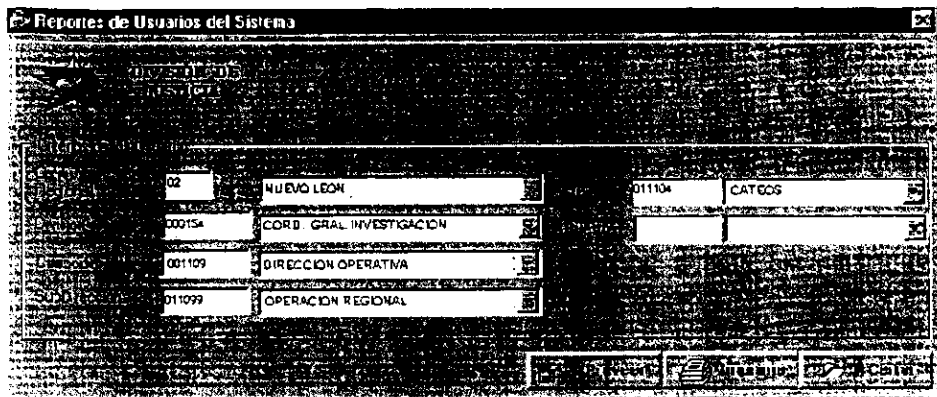
Dentro de éste menú el **Administrador** podrá consultar en pantalla previo a su impresión todos aquellos reportes de Usuarios que se encuentren en algún Sistema. Todos los reportes se podrán generar dados diferentes criterios de búsqueda permitidos al administrador.



Pantalla del segundo menú. **Consultas y Reportes**. En ella también están las cuatro opciones Jurídicas más importantes de la Institución.

Opción **Reporte de Usuarios**.

Dentro de ésta opción el **Administrador** podrá consultar los **Usuarios** con los criterios de búsqueda proporcionados, éstos criterios son: **Centro Regional**, **División**, **Dirección**, **Subdirección**, **Área** y por **Puesto** como se muestra en la siguiente pantalla.



Reportes de Usuarios.

Opción Reporte de Sistema.

En esta opción el Sistema permite obtener el Reporte de los Usuarios existentes dentro de cada Sistema, aquí también se permite visualizarlo en pantalla antes de imprimirlo, y los criterios para generar éste reporte son:

a) Criterios por datos de Usuarios.

Por centro regional, por División, por Dirección, por Subdirección, por Área, por Puesto.

b) Criterios por datos del Sistema.

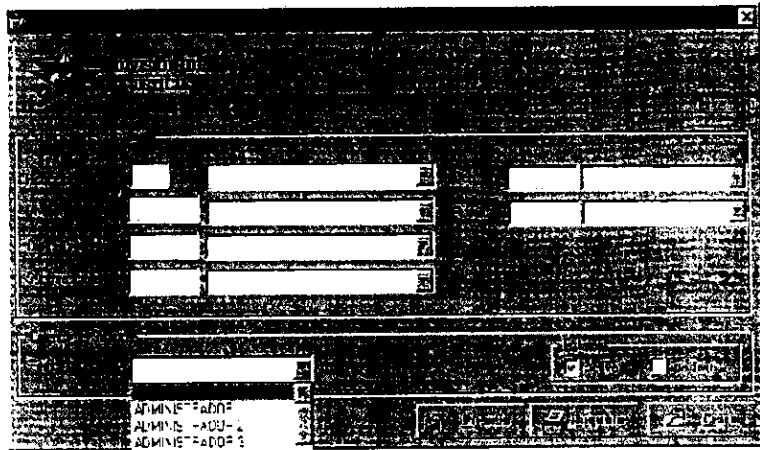
Por Perfil, y por estatus de Usuarios.

Un ejemplo de la pantalla que permite generar el Reporte de Averiguaciones Previas es el que a continuación se muestra:

a) Criterios por datos de Usuarios

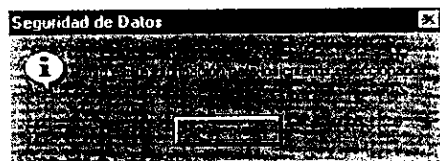


b) Criterios por datos del Sistema



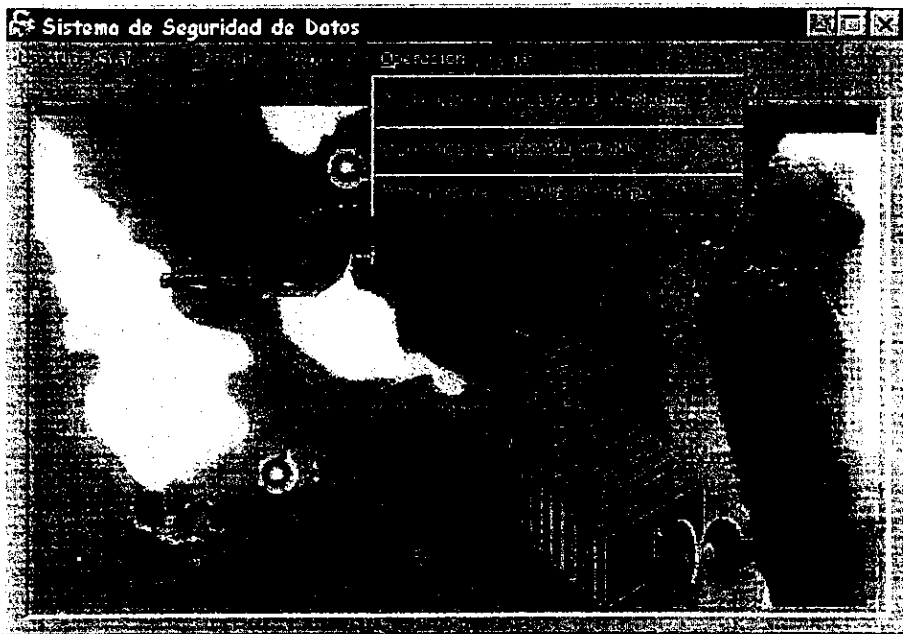
Reporte del Sistema de Averiguaciones Previas.

En el caso de que el sistema no encuentre información que cumpla con los criterios seleccionados en cada opción, no se podrá generar el reporte y notificará al administrador mediante el siguiente mensaje:



3) Menú Operación.

Dentro de éste menú se encuentran acciones que únicamente uno de los dos grupos definidos por los Usuarios podrá acceder al grupo denominado **Master**. Este grupo podrá realizar acciones críticas entre las que se encuentran **Mantenimiento a Administradores**, **Mantenimiento de Grupos** y **Mantenimiento a Catálogos** del sistema.



Tercer pantalla del menú principal. Operación. En ella están las tres opciones de Mantenimiento del sistema.

Para realizar la Alta de un Administrador se presentará la siguiente pantalla, en la cual hay que capturar la información que se solicita.

Mantenimiento a Administradores del sistema.

De igual forma para dar de Alta Grupos en los cuales se incluirán los Usuarios propios de éste Sistema de Seguridad de Datos se presentará la siguiente pantalla:

Pantalla del Menú de Operación.
Mantenimiento a Grupos del sistema

Dentro del Sistema existen ciertos campos que ya están predeterminados y únicamente es cuestión de que el administrador lo seleccione de la lista de opciones que aparecen en la pantalla, para ello la base de datos cuenta con catálogos que el mismo administrador deberá ir poblando según las necesidades de la División Jurídica. Este mantenimiento a catálogos es posible mediante la siguiente pantalla:

CATALOGO DE CENTROS REGIONALES	
ES LA DIVISION POR ESTADOS DE LA REPUBLICA MEXICANA	
01	JALISCO
02	MONTERREY
03	DISTRITO FEDERAL

Menú de Operación.
Mantenimiento a Catálogos del sistema.

CONCLUSIONES.

Con la implantación de la Base de Datos para la Administración y Control en el Acceso en los Sistemas Jurídicos se podrá eliminar uno de los puntos más graves que se le atañe a las instituciones de impartición de justicia que es la confiabilidad.

Se logrará tener registrados todos y cada uno de los movimientos que efectúe tanto los Usuarios como los Administradores del sistema, reflejando con ello un transparente funcionamiento tanto en el sistema jurídico como en la confiabilidad de la información.

Un punto importante a destacar es el que con los registros detallados que obtiene el Sistema son de gran ayuda en el combate de la fuga de información que se realiza cotidianamente en la institución, ya que se tiene registro exacto de todos y cada uno de los usuarios y administradores que pudieran caer en un delito como ese.

Al tenerse un control pleno de los cuatro sistemas claves para la Institución Jurídica, se logrará tener una información exacta, confiable y libre de poder ser usada con otros fines ilícitos.

FUENTES DE INFORMACIÓN

BIBLIOGRAFIA

- ◆ **GONZALEZ, Alfonso.** SQL Server: programación y administración. México: Ed. Alfaomega. 1999.
- ◆ **PIATTINI, Mario G.** Fundamentos y modelos de bases de datos. México: Ed. Alfaomega. 1998.
- ◆ **HANSEN, Gary W. y HANSEN JamesV.** Diseño y Gestión de bases de datos. Madrid: Ed. Prentice Hall. Segunda Edición. 1997.
- ◆ **ELMASRI Ramez.** Sistemas de bases de datos. Conceptos fundamentales.- México: Ed. Addison Wesley Iberoamericana. 1997.
- ◆ **DATE, C.J.** Introducción a los sistemas de bases de datos. México: Ed. Addison Wesley Iberoamericana. 1996.
- ◆ **GARDARIN, Georges.** Dominar las bases de datos: modelos y lenguajes. Barcelona: Ed. Eyrolles. 1994.
- ◆ **MARTIN, James; Odell, James J.** Análisis y Diseño Orientado a Objetos, México, Prentice Hall Hispanoamericana S.A. 1994.
- ◆ **DEAKIN Rose** Bases de Datos. Uso y administración en centros de cómputo. México: Ed. Trillas. 1988.
- ◆ **DEEN, S.M.** Fundamentos de los sistemas de bases de datos. Barcelona: Ed. Gustavo Gill S.A. 1987.
- ◆ **DELOBEL, Claude.** Bases de datos y sistemas relacionales. Barcelona: Ed. Ediciones Omega. 1987.
- ◆ **KRUGLINSKI, David.** Sistemas de administración de bases de datos. México: Ed. MacGraw-Hill. 1985.
- ◆ **CAMPDERRICH, Benet.** Técnicas de Bases de datos. Barcelona: Ed. Editores técnicos y asociados S.A. 1984.

FUENTES DE INFORMACIÓN

INTERNET

- 1. Databases and the Internet w3c2.com.au/steve/is3.html
- 2. Databases and the Internet www.dbmsmac.com/f19512.html
- 3. Modelos de Datos www.white/dbmdlref.htm
- 4. SGBD <http://www.inf.udec.cl/~revista>
- 5. Cliente/Servidor <ftp://docum/white/spanish/csarti.pdf>

GLOSARIO.

A

Algoritmo. Procedimiento matemático o lógico para realizar un cálculo o para resolver un problema. Sucesión de operaciones elementales, perfectamente especificadas y ordenadas, que sirven para hacer algo preciso.

Análisis del sistema. Fase de la metodología de desarrollo de sistemas en la que se definen objetivos, funciones, pantallas, menús, modelos de datos, arquitecturas y, en general, todas las cuestiones necesarias antes de proceder al diseño de programas para el lenguaje que se haya elegido.

ANSI X.12. Comité de Normalización EDI para ANSI.

ANSI. *American National Standards Institute.* Instituto Nacional Americano para la Estandarización. Organismo oficial dedicado a fomentar la adopción de normativas en materia de informática, comunicaciones, etc.

Aplicación. Programa informático que proporciona servicios de alto nivel al usuario, generalmente utilizando otros programas más básicos que se sitúan por debajo.

Arbol. Estructura de representación de la información que consiste en un único registro "padre" del que dependen cero o más registros "hijos" que, a su vez, pueden dar origen a nuevos subárboles.

Arquitectura de información. Conjunto de conceptos que definen cuál será el estilo que una organización va a dar a sus sistemas informáticos, afectando a temas como la arquitectura de red, el modo de proceso, y la distribución de capacidad de proceso y almacenamiento.

ASCII. *American Standard Code for Information Interchange.* Código estándar americano para intercambio de la información. Esquema normalizado de codificación de caracteres introducido en 1.963 y muy utilizado en muchas máquinas. Sistema de codificación de caracteres alfanuméricos en 7 bits para la operación interna del computador y su comunicación con los periféricos. Este sistema, promovido por el ANSI (*American National Standard Institute*), es ampliamente utilizado por ordenadores personales, estaciones de trabajo y miniordenadores.

Atributo de fichero. Clasificación de acceso a archivos que permite leer o buscar un fichero. Los atributos comunes son leer/escribir, sólo lectura, archivo y oculto. Característica de un fichero normalizada. Los atributos se clasifican en grupos de atributos. (Los ficheros de sistema de ficheros reales también tienen atributos, pero no están normalizados).

Atributo. Propiedad de un objeto o entidad.

B

Back-end (herramientas). Herramienta CASE de bajo nivel dirigida a las últimas fases del desarrollo: construcción e implantación.

Backup. Véase Copia de Seguridad.

Base de datos. *Data base.* Conjunto de datos no redundantes, almacenados en un soporte informático, organizados de forma independiente de su utilización y accesibles simultáneamente por distintos usuarios y aplicaciones. La diferencia de una BD respecto a otro sistema de almacenamiento de datos es que estos se almacenan en la BD de forma que cumplen tres requisitos básicos: no redundancia, independencia y concurrencia.

Bases de Datos Documentales. Son bases de datos textuales con características de búsqueda documental asociadas: diccionario, tesauros, operadores de proximidad, operadores booleanos, truncamientos, etc.

C

C. Lenguaje de programación estructurado, de aplicación general.

C++. Lenguaje C orientado a objetos.

CASE. *Computer Aided Software Engineering.* Ingeniería de *software* asistida por ordenador. Son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. La principal

GLOSARIO.

ventaja de la utilización de una herramienta CASE es la mejora de la calidad de los desarrollos realizados y en segundo término el aumento de la productividad. El objetivo final de la tecnología CASE es el de producir *software* de forma automática según los diseños construidos mediante herramientas CASE.

Ciclo de vida. Conjunto organizado de actividades y fases, tanto técnicas como de gestión que es necesario desarrollar a lo largo de la vida del sistema, desde que se decide su necesidad hasta que el sistema deja de ser utilizado. Según METRICA versión 2.1 estas fases son: planificación, análisis, diseño, construcción e implantación.

Clase. Modelo de objetos o entidades. Todos los objetos de una clase comparten las mismas propiedades, mensajes y métodos de esa clase.

Clave. Campos de datos que identifican unívocamente un registro en un archivo.

Cliente/Servidor. Arquitectura de sistemas de información en la que los procesos de una aplicación se dividen en componentes que se pueden ejecutar en máquinas diferentes. Modo de funcionamiento de una aplicación en la que se diferencian dos tipos de procesos y su soporte se asigna a plataformas diferentes.

Codificación. Transformación de un mensaje en forma codificada, es decir, especificación para la asignación unívoca de los caracteres de un repertorio (alfabeto, juego de caracteres) a los de otro repertorio. || Conversión de un valor analógico en una señal digital según un código prefijado.

Código binario. Código en el que los elementos se representan solamente por los valores "1" y "0". Es el código empleado principalmente dentro de los circuitos de los equipos físicos.

Código. Cada una de las secuencias de caracteres que transforman los elementos de un repertorio en otro.

Compilador. Programa que genera una copia del programa fuente en lenguaje máquina para que el ordenador pueda procesarlo directamente.

Confidencialidad: Consiste en proteger la BD contra accesos no autorizados. Debe asegurarse no sólo que los usuarios no autorizados no consigan acceso a la BD sino, también, que los usuarios legítimos acceden sólo a los datos autorizados.

CORBA (Common Object Broker Architecture, o *Arquitectura común de gestores de solicitudes de objetos*), del Object Management Group (OMG).

D

Data Mining (Minería de Datos). Proceso que permite el descubrimiento y cuantificación de relaciones predictivas en los datos, permitiendo transformar la información disponible en conocimiento útil de negocio.

Data Warehouse. Almacén de datos, integrado en una estructura consistente, organizado por temas, con valores históricos e información no volátil.

Datos. Hechos aislados.

DDL. Data Definition Language. Lenguaje de definición de datos. Se utiliza para la definición de la base de datos y de los elementos que contiene (tablas, relaciones...) y para su mantenimiento. Es propio de cada SGBD. Permite definir entidades, identificadores (claves), atributos, interrelaciones, autorizaciones de acceso, restricciones de integridad, etc. A nivel interno, facilita la definición del espacio físico, longitud de los campos, representación de los datos (binario, alfanumérico, etc.), caminos de acceso (punteros, índices...), etc.

Decodificación. Conversión de un valor digital en una señal analógica. || Proceso de reconversión de un mensaje codificado al mensaje que dio lugar a la codificación.

Diccionario de datos. Descripción lógica de los datos para el usuario. Reúne la información sobre los datos almacenados en la BD (descripciones, significado, estructuras, consideraciones de seguridad, edición y uso de las aplicaciones, etc.).

Directorio de datos. Es un subsistema del SGBD que describe dónde y cómo se almacenan los datos en la BD (modo de acceso y características físicas de los mismos).

Directorio de ficheros. Concepto que utilizan algunos sistemas de almacenamiento de ficheros como medio de organización. Un directorio es un almacén de ficheros que puede además contener

GLOSARIO.

otros almacenes de ficheros a los que proporciona acceso (un directorio está implementado como un fichero que contiene nombres y direcciones de ficheros)

DML (*Data Manipulation Language*): Lenguaje de manipulación de datos, se utiliza para la actualización y consulta de los datos almacenados en la base de datos. Permite añadir, seleccionar, suprimir o modificar los datos de la BD respetando las reglas establecidas por el **DDL**.

Dominio. En administración de bases de datos, todos los valores posibles que puede contener un campo en particular para cada registro en el archivo. || En comunicaciones, todos los recursos que están bajo el control de un sólo sistema de computación. || En dispositivos de almacenamiento magnético, grupo de moléculas que constituyen un *bit*.

E

ECMA. *European Computer Manufacturers Association*. Asociación Europea de Fabricantes de Ordenadores. Coordina a los fabricantes en las organizaciones internacionales de normalización.

F

Front-end (herramientas). Herramientas de planificación y definición de requisitos, herramientas de análisis y diseño y herramientas de modelización y generación de prototipos. Véase Terminal.

G

Grafo. Estructura matemática formada por elementos llamados nodos y arcos. Un arco es una relación (representada gráficamente por una línea) que puede existir entre cada pareja de nodos.

H

Herencia. Notación ligada a los objetos estructurados que permite a los objetos relacionados jerárquicamente heredar atributos.

Hijo. Un registro "subordinado" en una interrelación jerárquica.

I

Inconsistencia. El contenido de una base de datos es inconsistente si dos datos que deberían ser iguales no lo son. Por ejemplo, un empleado aparece en una tabla como activo y en otra como jubilado.

Indexación. Proceso automático de generación de índices.

Información. Datos organizados o resumidos.

Integridad: Mantener la integridad de una base de datos es asegurarse de que los datos que contiene son correctos, evitando datos inconsistentes o erróneos de cualquier otro tipo. Por ejemplo, que un empleado aparezca como perteneciente a un departamento que no existe en la tabla correspondiente.

Interfaz Gráfica de Usuario (GUI). Pantallas y funciones que proporcionan al usuario final un medio gráfico para tener acceso al sistema de computación.

M

Memoria. Zonas de un ordenador o periférico donde se almacenan temporalmente los datos y los programas con los cuales se trabaja en un determinado instante.

Metadatos. Datos acerca de los datos. Fichero que debe existir en todo Data Warehouse, que incluye todas las características de los datos que se almacenan en él y las transformaciones que sufren del entorno operacional al Data Warehouse: Campos que lo componen, estructura, procedencia, proceso de transformación, acumulación, forma de cálculo, etc. || Datos en el

GLOSARIO.

diccionario de datos que describen la base de datos

Middleware. Equipo lógico intermedio que permite la comunicación entre dos módulos de *software*. Para compartir datos, los dos módulos de *software* no necesitan saber como comunicarse entre ellos, sino como comunicarse con el módulo de *middleware*. El *middleware* debe ser capaz de traducir la información de una aplicación y pasársela a otra. || En términos de programación orientada a objeto, el módulo *middleware* identifica diferentes objetos y conoce qué propiedades tienen asociadas, por lo que puede responder a peticiones referentes a los mismos.

Modelo conceptual de datos. Diseño previo de una base de datos en el que se realiza la primera aproximación.

Modelo de datos. Un conjunto de reglas que especifican a alto nivel la estructura de la base de datos. Según el modelo utilizado, los mismos datos se almacenarán de forma muy distinta. || Un método conceptual para estructurar los datos.

Modelo jerárquico. Un modelo de datos que resume que todas las interrelaciones entre los datos pueden estructurarse como jerarquías.

Modelo relacional. Propuesto por Codd a finales de los sesenta. Introduce la teoría de las relaciones en el campo de las BD. En este modelo, los datos se estructuran en tablas manteniendo la independencia de esta estructura lógica respecto al modo de almacenamiento u otras características físicas. Las tablas se manejan mediante operaciones de la teoría de conjuntos y el álgebra relacional.

N

Normalización. Según el modelo relacional, las tablas deben definirse siguiendo una serie de reglas precisas para asegurarse que no se producirán anomalías en la actualización de la base de datos. Para ello, es habitual que se necesite descomponer las tablas iniciales en otras más simplificadas que no presenten dichos problemas. Este proceso es lo que se conoce como normalización y es un método formalizado con diferentes niveles, a cada uno de los cuales se le llama forma normal. || Acto de producción de normas y estándares.

O

OLAP (On Line Analytical Processing). Sistemas que se caracterizan por ser un análisis multidimensional de datos corporativos, soportando análisis del usuario y con posibilidades de navegación en base a la información a obtener.

OLTP. On Line Transaction Processing. Proceso transaccional en línea. Método de proceso continuo de transacciones.

P

Padre. Un registro "propietario" en una interrelación jerárquica.

Plataforma cliente/servidor. Una red local que consiste en computadores clientes, que reciben servicios de un computador servidor.

Proceso. Unidad ejecutable elemental que es manejada por el sistema operativo en su función de asignar tiempo de UCP. || En *workflow*, conjunto de tareas.

Programa. Conjunto de instrucciones escritas en un lenguaje específico que un ordenador sigue para realizar una tarea concreta.

Protocolo. Conjunto formal de convenciones que gobiernan el formato y control de datos. Conjunto de procedimientos o reglas para establecer y controlar transmisiones desde un dispositivo o proceso fuente a un dispositivo o proceso objeto.

Prototipo. Modelo operacional de prueba indicado para la evaluación de las especificaciones de un sistema de información, o para un mejor entendimiento de cómo los requerimientos son plasmados en la aplicación desarrollada.

Puntero. Una dirección física que identifica dónde puede encontrarse un registro sobre el disco.

GLOSARIO.

R

Recuperación: Su objetivo es proteger a la BD contra fallos (lógicos o físicos) que destruyan su contenido parcial o totalmente. Los SGBDs suelen incluir los llamados "ficheros de *log*" en los que se almacenan todos los cambios antes de almacenarlos en la BD, así como, marcas de comienzo y final de transacción. A partir de ellos, el SGBD puede decidir después de un fallo, si una transacción estaba terminada o no y, por tanto si hay que mantenerla o deshacerla.

Red. Una interrelación de datos en la cual un registro puede estar subordinado a registros de más de un archivo.

Redundancia: Se llama redundancia al hecho de que los mismos datos estén almacenados más de una vez en la base de datos. Las redundancias además de suponer un consumo de recursos de almacenamiento pueden llevar a situaciones en las que un dato se actualice en una de sus ubicaciones y en otra no y se pierda la integridad de la BD, por tanto deben evitarse.

Registro. Grupo de elementos de datos relacionados entre sí que son tratados como una sola unidad. Soporte interno (informático) de la estructura de la información a grabar. Su diseño responde a los datos contenidos en la información que se desea grabar.

Relación. Matemáticamente, una relación es un subconjunto del producto cartesiano de n dominios no necesariamente distintos. La teoría matemática de las relaciones, adaptada y modificada, fue la base del modelo relacional de bases de datos.

S

Servidor de bases de datos. Un programa que corre en un computador servidor para dar servicios de bases de datos a las máquinas clientes.

SGBD. Sistema de Gestión de Bases de Datos. Conjunto de programas que permite crear una base de datos, manipular la información que contiene y realizar todas las tareas de administración necesarias para mantenerla operativa.

Sistema abierto. El concepto de conectar una variedad de computadores con diferentes hardware y software para trabajar coordinadamente con el fin de lograr los objetivos del usuario.

Sistema de base de datos. Una base de datos, un sistema de gestión de base de datos, con el hardware y el personal apropiados.

Sistema de información para la gestión. Sistema que provee información para la gestión o la administración.

Sistema de información. Sistema automatizado que organiza los datos para producir información.

Sistema operativo. Conjunto de programas responsable de la explotación del ordenador, mejorar su nivel de rendimiento y gestionar los recursos del mismo. Conjunto de programas que gestionan y distribuyen los recursos del ordenador.

SQL (Structured Query Language). El SQL es un lenguaje de alto nivel, no procedural, normalizado que permite la consulta y actualización de los datos de BD relacionales. Se ha convertido en el estándar de acceso a BD relacionales. La primera versión se aprobó como norma ISO en 1987 y la segunda, conocida como SQL2 y vigente actualmente, en 1992. Actualmente se trabaja en la norma SQL3 que soportará bases de datos orientadas a objeto y bases de datos activas.

T

Trigger Desencadenante. Suceso que no es parte del proceso pero hace que este arranque.

Tupla. Fila de una tabla de un SGBD relacional. Representa una ocurrencia.