

22  
Zij



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

Arquitectura Cliente/Servidor en un  
sistema para la DGENP

TESIS PROFESIONAL  
Que para obtener el Título de  
INGENIERO EN COMPUTACION

P r e s e n t a n  
Javier Benítez Macías  
Mario Ibáñez Nájera

Director de Tesis:  
ING. LUIS G. CORDERO BORBOA



México, D. F.,

1996.

TESIS CON  
FALLA DE ORIGEN

TESIS CON  
FALLA DE ORIGEN



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.

**A la Universidad Nacional Autónoma de México:**

*Por ser la base de nuestra formación profesional y personal.*

**A la Dirección General de la Escuela Nacional Preparatoria:**

*Por el apoyo y las facilidades otorgadas para la realización de éste estudio.*

**Al Ing. Luis Cordero Borboa:**

*Por sus conocimientos y el apoyo brindado en el desarrollo de éste estudio.*

**Al Lic. Julio César Roldán Campos:**

*Por su gran apoyo y amistad invaluable.*

**A mis Padres:**

*Por su amor y apoyo, por estar conmigo en todo momento y por darme un ejemplo constante de superación.*

**A mis Hermanos:**

*Porque son una parte muy importante en mi vida y me motivan a ser mejor día a día.*

**A Elizabeth Rosales:**

*Por su amor, por compartir conmigo este esfuerzo, y por todas las cosas que hemos aprendido juntos.*

**A mis familiares y amigos:**

*Por contar siempre con su amistad, y por que de cada uno de ellos he aprendido algo valioso para mi. En especial a Mario por su dedicación y amistad.*

**Javier**



**A mis Padres:**

*Por que gracias a su gran apoyo y a su amor incomparable he podido salir adelante, motivándome para ser cada vez mejor.*

**A mis Hermanos:**

*Por su amor, comprensión y motivación para seguir esforzándome día a día.*

**A mis familiares y amigos:**

*Por contar siempre con su amistad. En especial a Javier por su colaboración para que éste trabajo fuera posible..*

**Mario**

<b>ÍNDICE</b>
---------------

<b>INTRODUCCIÓN</b> .....	i
<b>CAPÍTULO I CONCEPTOS GENERALES</b> .....	1
<b>I.1 Bases de Datos.</b> .....	3
1.1.1 ¿Qué es un Manejador de Bases de Datos?. .....	5
1.1.2 Modelos de Manejadores de Bases de Datos. ....	5
<b>I.2 Lenguajes de programación.</b> .....	9
<b>I.3 Tipos de arquitecturas.</b> .....	10
<b>I.4 Selección de la arquitectura.</b> .....	15
<b>I.5 Arquitectura Cliente/Servidor.</b> .....	16
<b>I.5.1 Fundamentos.</b> .....	16
<b>I.5.2 Ventajas y Desventajas.</b> .....	23
<b>I.5.3 Seguridad.</b> .....	24
 <b>CAPÍTULO II COMPONENTES DE LAS APLICACIONES C/S</b> .....	 29
<b>II.1 El cliente.</b> .....	32
<b>II.1.1 El papel del cliente.</b> .....	34
<b>II.1.2 Servicios del cliente.</b> .....	36
<b>II.1.3 Petición para el servicio.</b> .....	38
<b>II.1.4 Llamadas de procedimiento remota.</b> .....	38
<b>II.1.4.1 Servicios.</b> .....	39
<b>II.1.5 Plataformas.</b> .....	40
<b>II.1.5.1 Principales plataformas.</b> .....	42
<b>II.1.5.1.1 Novell Netware.</b> .....	42
<b>II.1.5.1.2 Windows NT.</b> .....	43
<b>II.1.5.1.3 UNIX.</b> .....	43
<b>II.1.5.1.4 OS/2.</b> .....	44
<b>II.1.5.1.5 LAN Manager.</b> .....	45
<b>II.1.6 Protocolos de la comunicación.</b> .....	45
<b>II.1.6.1 NetBios (Network Basic Input/Output System).</b> .....	45
<b>II.1.6.2 IBM APPC (Application Program-to-Program Communication).</b> ..	46
<b>II.1.6.3 TCP/IP (Transmission Control Protocol/Internet Protocol).</b> .....	46
<b>II.1.6.4 Sockets.</b> .....	48
<b>II.1.7 Interfaces Gráficas de Usuario.</b> .....	49
<b>II.1.8 Las aplicaciones.</b> .....	50
<b>II.1.9 Seguridad.</b> .....	51
<b>II.1.9.1 Proceso de autenticación.</b> .....	51
<b>II.1.9.2 Kerberos.</b> .....	52
<b>II.1.9.3 Una sola conexión.</b> .....	52
<b>II.1.9.4 Confidencialidad.</b> .....	53

II.1.9.5 Sistemas operativos y seguridad. ....	54
II.2 Programa de conexión. ....	56
II.3 Servidores. ....	58
II.3.1 Red de cómputo. ....	59
II.3.2 Distribución entre el cliente y el servidor. ....	59
II.3.3 Interfaz de usuario (Front-End)- Sistema de fondo (Back-End). ....	60
II.3.4 Consideraciones para la implementación de servidores. ....	61
<b>CAPÍTULO III DISEÑO DE LA BASE DE DATOS .....</b>	<b>63</b>
III.1 Etapas del proceso de diseño. ....	67
III.2 Modelo Entidad-Relación. ....	69
III.3 Diseño del modelo Entidad-Relación. ....	71
III.4 Diseño del esquema lógico. ....	74
III.5 Conversión del esquema lógico a una base de datos física. ....	75
III.6 Sistemas Manejadores de Bases de Datos. ....	76
III.6.1 Oracle7. ....	80
III.6.2. Ingres. ....	83
III.6.3. Sybase. ....	87
<b>CAPÍTULO IV ELECCIÓN DEL DBMS Y LA INTERFAZ DE USUARIO ----</b>	<b>93</b>
IV.1 Elección del manejador de base de datos. ....	95
IV.1.1 Criterios de evaluación. ....	95
IV.1.1.1 Estructura y Arquitectura. ....	96
IV.1.1.2 Funcionalidad general. ....	100
IV.1.1.3 Funcionalidad distribuida. ....	102
IV.1.1.4 Desempeño. ....	104
IV.1.1.5 CPU. ....	105
IV.1.1.6 Memoria. ....	107
IV.1.1.7 Entradas/Salidas a disco. ....	108
IV.1.1.8 Ambiente de desarrollo. ....	109
IV.1.1.9 Utilerías. ....	110
IV.1.2 Elección de la herramienta. ....	111
IV.2 Elección de la herramienta visual para desarrollo cliente/servidor. ....	122
IV.2.1 Productos evaluados. ....	122
IV.2.2 Criterios de evaluación. ....	123
IV.2.3 Como trabajan las interfaces de usuario. ....	125
<b>CAPÍTULO V ESTUDIO PRELIMINAR DEL SISTEMA .....</b>	<b>131</b>
V.1 Definición del problema. ....	134
V.2 Alcance del sistema. ....	135
V.2.1 Inscripciones. ....	136
V.2.2 Reinscripciones. ....	137

V.2.3 Bases definitivas voluntarias. ....	138
V.2.4 Exámenes extraordinarios. ....	138
V.2.4.1 Hasta dos exámenes extraordinarios. ....	138
V.2.4.2 Mas de dos exámenes extraordinarios. ....	139
V.2.5 Control de actas. ....	139
V.2.6 Seguro facultativo. ....	140
V.2.7 Registra sanciones académicas. ....	140
V.2.8 Emite documentos. ....	141
V.2.9 Emite estadísticas. ....	141
V.3 Justificación de la metodología. ....	142
V.3.1 Análisis de metodologías. ....	142
V.3.1.1 Metodología Jackson. ....	142
V.3.1.2 Análisis orientado a objetos. ....	143
V.3.1.3 Prototipos. ....	147
V.3.1.4 Análisis Estructurado. ....	149
V.3.2 Metodología Seleccionada. ....	150
<b>CAPÍTULO VI ANÁLISIS Y PROPUESTA DEL SISTEMA</b> .....	<b>153</b>
VI.1 Modelo ambiental. ....	156
VI.1.1 Objetivo del sistema. ....	156
VI.1.2 Diagrama de contexto. ....	157
VI.1.3 Lista de eventos. ....	159
VI.2 Modelo de comportamiento. ....	161
VI.2.1 Diagrama de flujo de datos. ....	161
VI.2.2 Miniespecificaciones. ....	172
VI.2.3 Diccionario de datos. ....	177
VI.3 Modelo de información. ....	200
VI.3.1 Diagrama de Entidad-Relación (DER). ....	203
VI.4 Entidades, Relaciones y Atributos. ....	204
VI.4.1 Entidades y sus atributos. ....	204
VI.4.2 Relaciones y sus atributos. ....	207
VI.4.3 Descripción de las entidades y sus atributos. ....	211
VI.4.4 Descripción de las relaciones y sus atributos. ....	226
VI.5 Diagramas de transición de estados. ....	242
VI.6 Diagramas de estructura. ....	247
VI.6.1 Especificación de módulos. ....	250
VI.7 Arquitectura de Conexiones. ....	251
CONCLUSIONES. ....	255
BIBLIOGRAFÍA. ....	257

## INTRODUCCIÓN

El área de la computación ha tenido una evolución muy significativa, debido a la competitividad que existe en el mercado y a las necesidades del usuario que día a día aumentan. En los últimos años el aspecto del crecimiento ha sido demasiado marcado por la integración de nuevas tecnologías para la creación tanto de programas ("Software" en inglés) como de equipo ("Hardware" en inglés). Todo esto ha ocasionado que la computación incursione en otras áreas de trabajo y que los resultados proporcionados sean extremadamente satisfactorios, y con esto crear nuevas expectativas de desarrollo. Actualmente el mundo de la informática se encuentra presente en todas partes y en nuestra vida diaria.

Para muchas instituciones, los sistemas computacionales de información son muy indispensables para la automatización de sus actividades.

En la Dirección General de la Escuela Nacional Preparatoria (DGENP) de la UNAM, surge el interés y la necesidad de cambiar a otro ambiente y a otra plataforma de trabajo en cuanto a sus sistemas se refiere; es por eso que en ésta Tesis se presenta una propuesta de conversión de los sistemas actuales que trabajan bajo una arquitectura servidor de archivos y en una red local, a una arquitectura cliente/servidor (C/S) bajo una plataforma Unix, con lo cual se tendrá mayor confiabilidad y velocidad en el tráfico de información.

En el capítulo 1 se tratan las arquitecturas de bases de datos (BD), tipos de arquitecturas y como seleccionar una de ellas. También se habla más a detalle de lo que es la arquitectura C/S, ventajas y desventajas y el por que de usar una arquitectura de este tipo.

En el capítulo 2 se engloban las partes que conforman una arquitectura C/S. Los servidores, los clientes y capas intermedias o programas de conexión ("Middleware" en inglés), describiendo tipos, ventajas, desventajas y comparaciones.

En el capítulo 3 se trata el diseño de las bases de datos, manejando los puntos más importantes para ello y haciendo uso del modelo Entidad-Relación.

En el capítulo 4 se lleva acabo esencialmente la elección de las herramientas de que se hará uso, tomando en consideración que cumplan con las necesidades requeridas.

En el capítulo 5 se analiza el estado de los sistemas de la DGENP en la actualidad, la problemática que existe y el alcance del sistema, así como una elección de la metodología usada para el análisis y diseño del sistema.

En el capítulo 6 se realiza el diseño del sistema enfocándolo bajo el concepto C/S.

El Sistema Manejador de Bases de Datos (DBMS) que se ocupa en este desarrollo es SYBASE ya que como se verá en el estudio comparativo SYBASE es el servidor de bases de datos más adecuado para el ambiente que se va a utilizar, UNIX, y por ofrecer la mayor interoperabilidad tanto con otros servidores de bases de datos como con diversos programas de usuario final ("front-end" en inglés) y herramientas de desarrollo, en su gran mayoría no propietarias, lo que le permitirá adaptarse al entorno universitario donde la diversidad de software es común.

Cabe hacer mención que este trabajo de Tesis es una propuesta escrita de migración a una nueva arquitectura y que no se realizará físicamente debido a que no se cuenta hasta el momento con el equipo necesario. Deseamos y esperamos que ésta aportación tenga un máximo beneficio tanto para la DGENP, como para cualquier otra dependencia de la UNAM.

# **CAPÍTULO I**

## ***CONCEPTOS GENERALES***

## CAPÍTULO I CONCEPTOS GENERALES

### OBJETIVO DEL CAPÍTULO

En éste primer capítulo se presentarán conceptos generales relacionados con los temas que involucran a ésta tesis y serán de gran ayuda para la comprensión de los capítulos posteriores.

### INTRODUCCIÓN

Un sistema de bases de datos permite operar un conjunto de datos o información relevante para el trabajo o para fines personales. También proporcionan la capacidad de almacenar y mantener esa información en un solo lugar o en forma distribuida.

Un sistema de bases de datos consiste en dos partes: el Sistema Manejador de Bases de Datos ("DBMS" por sus siglas en inglés) que es el programa que organiza y mantiene los datos, y el programa de aplicación, el cual permite consultar, insertar y actualizar la información almacenada.

Una arquitectura Cliente/servidor incrementa el poder de procesamiento al separar el manejador de la base de datos de la aplicación. La aplicación se ejecuta en una o más estaciones de trabajo, que comúnmente son computadoras personales ("PC" por sus siglas en inglés), y se comunican a través de una red con uno o más manejadores de bases de datos que se encuentran en otras computadoras.

### L1 BASES DE DATOS

Un **dato** es el elemento susceptible de una observación sin significado, es la unidad mínima de información.

**Información** es un conjunto ordenado de datos interrelacionados con el objeto de proporcionar un conocimiento para tomar decisiones.

Una **base de datos** puede definirse como una colección de datos interrelacionados y almacenados en conjunto sin redundancias innecesarias, cuya finalidad es la de servir a una aplicación o más, de la mejor manera posible. Los datos son almacenados de tal manera que resulten completamente independientes de los programas de las aplicaciones.

Una **entidad** puede ser un objeto tangible o intangible que tiene propiedades o características propias que son llamadas **atributos**.



Una **relación** es la asociación de dos o más entidades y puede ser:

- **Uno a uno.** En una relación de este tipo un elemento en A está asociado sólo con un elemento en B, y un elemento en B está asociado sólo con un elemento en A.
- **Uno a muchos.** En una relación uno a muchos un elemento en A está asociado con varios elementos en B y un elemento en B puede estar asociado sólo con un elemento en A.
- **Muchos a uno.** En una relación muchos a uno un elemento en A está asociado sólo a un elemento en B y un elemento en B puede estar asociado a varios elementos en A.
- **Muchos a muchos.** En una relación muchos a muchos un elemento en A está asociado a varios elementos en B y un elemento en B está asociado a varios elementos en A.

**Independencia de los datos.** Se puede definir como "la capacidad de modificar una definición de esquema en un nivel sin afectar la definición del esquema en el nivel inmediato superior". [KOR86]

Implica que los programas de aplicación y los datos sean completamente independientes.

La independencia de datos la podemos dividir en independencia física e independencia lógica.

**Independencia física** es la habilidad de modificar el esquema físico sin causar ningún cambio en los programas de aplicación, las modificaciones en el esquema físico son necesarias cuando deseamos mejorar el desempeño del sistema. [KOR86]

**Independencia lógica** es la habilidad de modificar el esquema conceptual sin necesidad de modificar los programas de aplicación, las modificaciones al nivel conceptual son necesarias cuando la estructura lógica de la base de datos ha sido alterada. [KOR86]

**Integridad** implica el uso de procedimientos para salvaguardar la base de datos de alteraciones no válidas o de su destrucción con datos inconsistentes.

La **llave primaria** identifica en forma única a cada renglón de la tabla. Una llave primaria no permite valores nulos y no puede tener valores duplicados.

La **llave foránea** es una columna de una tabla que es llave primaria en alguna otra. Indican asociaciones entre tablas.

### 1.1.1 ¿Qué es un manejador de Bases de Datos?

Un Sistema Manejador de Bases de Datos es una colección de archivos interrelacionados y un conjunto de programas que son utilizados por diferentes usuarios para acceder o modificar dichos archivos. El principal propósito de un manejador de bases de datos es proporcionar a los usuarios el acceso a los datos de manera eficiente y transparente, es decir completamente independiente a la forma de almacenamiento. [KOR86]

El manejador de base de datos es el responsable de proporcionar, al usuario, los servicios necesarios para manipular los datos.

Asimismo, el manejador de bases de datos debe de proveer algún tipo de integridad de los datos para evitar que éstos sean corrompidos. Esta característica es muy importante en bases de datos multiusuarios, en las cuales uno o más usuarios pueden actualizar un dato al mismo tiempo, por lo que el manejador de base de datos debe verificar que estos cambios se lleven a cabo ordenadamente.

Una base de datos puede almacenar cualquier tipo de información pero para que sea útil, los datos deben ser almacenados de acuerdo a su dominio. Un dominio es una categoría y tipo de dato que es guardado en un campo en particular.

Un manejador de base de datos debe proveer los siguientes servicios:

- La definición de datos.
- El mantenimiento de datos, en donde cada campo contiene información particular que describe al registro.
- La manipulación de datos permite al usuario insertar, modificar, borrar y ordenar datos.
- Despliegue de información, permite presentar al usuario los datos de diferentes maneras.
- La integridad de los datos provee diferentes métodos para evitar que la información sea corrompida

### 1.1.2 Modelos de Manejadores de Bases de Datos

Los sistemas manejadores de bases de datos pueden ser agrupados dentro de cuatro diferentes modelos: el Modelo Manejador de Archivos, el Modelo de Bases de Datos Jerárquicas, el Modelo de Bases de Datos Red y el Modelo de Bases de Datos Relacional. Cada modelo es una descripción conceptual de como funciona la base de datos. Describen, específicamente, cómo los datos son presentados al usuario y al programador y como se llevan a cabo los accesos. Asimismo, describen las relaciones entre las diferentes entidades.

Estos modelos no describen la manera de como los datos son almacenados en el disco. Estos detalles son responsabilidad de los diseñadores de los manejadores de bases de datos.

### **Modelo Manejador de Archivos**

En este modelo cada campo es almacenado secuencialmente en el disco en uno o varios archivos. Para realizar la búsqueda de un dato en particular, será necesario que se realice desde el principio del archivo hasta encontrar el registro deseado, por lo que se vuelve lenta la localización de la información.

Este modelo fue el primero en utilizarse para almacenar información en bases de datos computarizadas. La ventaja que tiene este modelo es su sencillez.

Dentro de las desventajas de este modelo tenemos que no existe ninguna indicación de las relaciones entre varios elementos dentro del almacenamiento secuencial. Asimismo, crea ciertos problemas a la integridad de los datos; los valores de los datos tienen que ser verificados por la aplicación antes de ser almacenados en el disco. La misma base de datos puede ser *accesada* por diferentes aplicaciones, y cada una puede tener diferentes valores para un mismo campo. Las diferentes aplicaciones tienen que ser controladas manualmente para asegurar que la definición de cada campo sea la adecuada respecto al dominio.

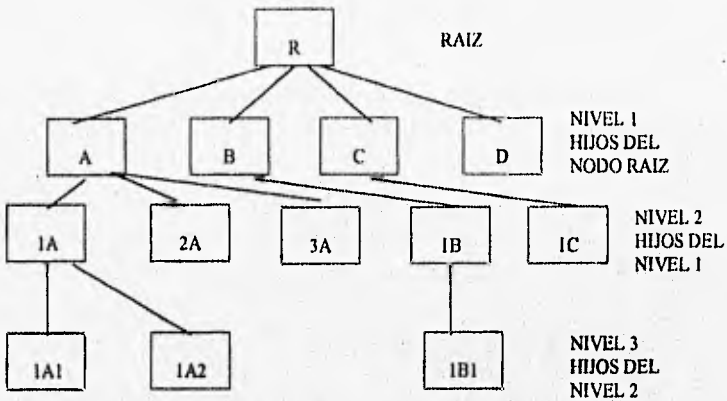
La única manera de ordenar los datos es leyendo todo el archivo y reescribiéndolo en el nuevo orden. Esto puede ser resuelto con el uso de un archivo de índices, el cual contiene apuntadores a cada registro de la base de datos.

El sistema manejador de archivos no permite que se modifique fácilmente la estructura de la base de datos.

### **Modelo de Bases de Datos Jerárquicas**

En este modelo los datos son organizados en una estructura de árbol, la cual comienza desde una raíz. Cada clase de dato puede ser localizado en diferentes niveles a partir de una rama cuyo origen es la raíz. En este tipo de estructura cada nivel es llamado nodo, en caso de que sea el último, es decir que dicha rama no tenga un nodo inferior es llamado hoja.

La estructura jerárquica permite que las búsquedas de datos sean más fáciles y rápidas. El manejador de base de datos no tiene que buscar en todo el archivo. Deberá examinar el requerimiento de la búsqueda y realizarla a partir de la raíz hacia una rama y los niveles inferiores. Un índice puede ser utilizado en este modelo para incrementar la velocidad de las búsquedas y puede ser creado en una clase particular de datos (nivel).



Fuente: [SAL93]

En un modelo jerárquico existe sólo un nodo llamado raíz, los apuntadores descienden desde la raíz a los nodos del nivel 1, que es donde realmente empiezan los datos.

La estructura física de los datos en el disco es completamente independiente del modelo jerárquico.

Este modelo facilita el agregar nuevos campos en cualquier nivel de la estructura, el manejador de base de datos sólo tendrá que cambiar el apuntador final para que apunte al nodo que se desea insertar.

La relación entre padres e hijos sólo podrá ser modificada si se rediseña toda la estructura. Otro problema de este modelo tan rígido es que no es fácil cambiar la definición de los niveles (clases de datos). Asimismo, no facilita la definición de relaciones de muchos a muchos.

### Modelo de Bases de Datos de Red

El modelo de bases de datos red consiste en una colección de registros conectados entre sí a través de ligas. Cada registro es un conjunto de campos (atributos) y una liga es una asociación entre dos registros. [KOR86]

El nombre de red no tiene nada que ver con el medio físico en el cual, actualmente corren las bases de datos. Este modelo describe conceptualmente las relaciones muchos a muchos. Las relaciones entre los diferentes tipos de datos son referidas como un solo grupo

para distinguirlas de las relaciones tan estrictas de padres-hijos definidas en el modelo jerárquico.

La flexibilidad del modelo red permite mostrar las relaciones de muchos a muchos. Estas interrelaciones entre los diferentes conjuntos pueden ser tan complejas que se dificulte su representación gráfica.

El diseño inicial de una base de datos bajo este modelo es fundamental, es decir que una vez que ha sido creado, cualquier cambio en algún grupo de datos implica realizar una nueva estructura.

### **Modelo de Bases de Datos Relacionales**

En 1969, el Dr. E. F. Codd publicó el primer artículo en el que se define un modelo de bases de datos basado en el concepto matemático de relaciones entre conjuntos.

En este modelo los datos son organizados en conjuntos lógicos y matemáticos en una estructura tabular, donde cada campo representa una columna de la tabla y cada registro un renglón.

Podemos tener varias tablas dentro de las cuales, una columna puede tener un nombre que se repite en otra tabla y son precisamente estas columnas repetidas las que nos permitieran relacionar las diferentes tablas.

El modelo relacional es completamente flexible en la descripción de relaciones entre varios grupos de datos. Asimismo, el cambiar la estructura de una tabla es tan sencillo como agregar o borrar una columna de la misma. Las tablas nuevas pueden crearse de tal manera que sean completamente independientes o bien como subconjuntos de las ya existentes.

La meta principal de este modelo es el preservar la integridad de los datos, por lo que se vuelve ideal para sistemas de procesamiento de transacciones, y para bases de datos con arquitectura cliente/servidor.

Un manejador de bases de datos relacional trata cada cambio (o conjunto de cambios) a los datos como una transacción, que es ejecutada en una copia temporal de la tabla que va a ser modificada. Los cambios no son efectuados hasta que el usuario o la aplicación realiza el commit de la transacción. En este modelo, el manejador de bases de datos controla los posibles conflictos que surjan de modificar algún dato.

---

## 1.2 LENGUAJES DE PROGRAMACIÓN

Los lenguajes utilizados para crear aplicaciones de bases de datos pueden ser agrupados en tres categorías: lenguajes estructurados, cláusulas SQL y otros lenguajes.

### Lenguajes Estructurados

En estos lenguajes cada procedimiento realiza una pequeña parte de la aplicación, como por ejemplo un procedimiento para realizar búsquedas, otro para actualizar datos, etc. Todos los procedimientos son ligados o unidos a través de un procedimiento general como puede ser un menú.

Las funciones de interfase de programación de aplicaciones generalmente están contenidas en librerías, las cuales son incluidas en la aplicación cuando es compilada.

Todos estos lenguajes son conocidos como lenguajes de tercera generación.

Los lenguajes estructurados que son específicos de algún manejador de bases de datos son conocidos como lenguajes de cuarta generación.

### SQL

El SQL (Structured Query Language) inicialmente fue desarrollado como un lenguaje para realizar accesos a una base de datos basada en el modelo relacional. La versión inicial de este lenguaje se conoció con SEQUEL y fue desarrollado por IBM como un lenguaje estándar que corría en computadoras grandes de la misma compañía. Posteriormente su nombre fue abreviado a SQL.

El SQL es el lenguaje más popular en arquitecturas cliente/servidor. Diferentes manejadores de bases de datos se pueden comunicar entre sí a través de SQL utilizando comandos para extraer y manipular datos.

Dentro de las diferentes versiones de SQL existen unas que son más estándar que otras, es decir que algunas contienen comandos o sintaxis que otras no pueden interpretar.

Es importante hacer notar que aunque el SQL es utilizado para acceder bases de datos generalmente de tipo relacional no quiere decir que cualquier base de datos relacional puede trabajar con SQL.

## Otros Lenguajes

Los más comunes de estos otros lenguajes son los de programación orientada a objetos. Estos representan una manera diferente de programación ya que los procesos o acciones son definidas en objetos en lugar de una serie de procedimientos.

Otro tipo de lenguaje es el de macro o escripts. Este lenguaje no es completamente de programación; son una lista de instrucciones que un usuario puede introducir en una aplicación para automatizar ciertas tareas.

Finalmente existe el QBE (Query by example) que no es estrictamente un lenguaje de programación, es una interfaz que le presenta al usuario una o más copias de las estructuras de las tablas que corresponden a las de la base de datos. El usuario selecciona las columnas que van a ser incluidas y establece las condiciones de búsqueda.

## 1.3 TIPOS DE ARQUITECTURAS

El tipo de sistemas de cómputo en el que las bases de datos corren puede clasificarse en cuatro diferentes categorías o plataformas: centralizadas, ambientes multiusuarios, computadoras personales, cliente/servidor y sistemas distribuidos. La arquitectura del manejador de base de datos no determina necesariamente el tipo del sistema de cómputo en el que la base de datos tiene que correr, aun cuando algunas arquitecturas son más comunes para ciertas plataformas que otras.

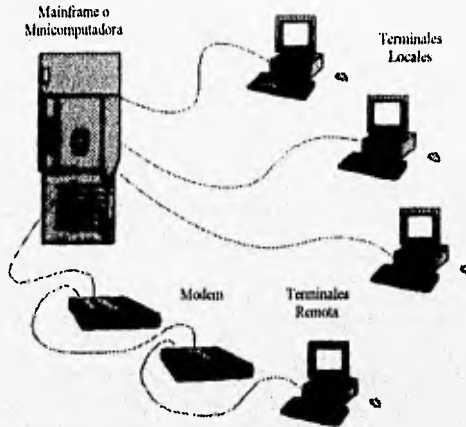
### Plataformas Centralizadas

En un sistema centralizado todos los programas corren en un "host" o computadora principal, incluyendo el manejador de base de datos, las aplicaciones que accesan a la base de datos y el software de comunicaciones que envían y reciben datos de las terminales de los usuarios.

Las aplicaciones y el manejador de base de datos corren en el mismo host, comunicándose a través de áreas compartidas de memoria y de tareas que son administradas por el sistema operativo del host. El manejador es el responsable de mover los datos de o al disco o sistema de almacenamiento utilizando los recursos proveídos por el sistema operativo.

Las ventajas principales de un sistema centralizado son, en primer lugar que el manejo de la seguridad también se encuentra centralizado y la habilidad de tener grandes cantidades de datos en dispositivos de almacenamiento. Asimismo, esta plataforma permite que existan numerosos usuarios simultáneos.

Las desventajas generalmente están relacionadas con los costos elevados de actualización, de mantenimiento de estos sistemas (pisos falsos, sistemas de enfriamiento por agua, sistemas de control de clima, etc.) y de un equipo de operadores y de programadores.



Fuente: [SAL93]

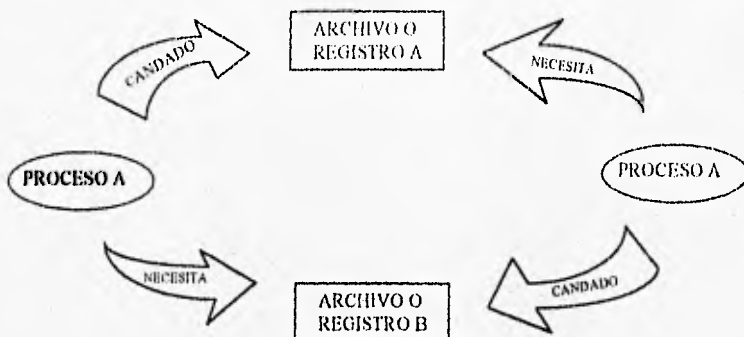
### **Ambientes Multiusuarios**

Para los sistemas de bases de datos multiusuarios los manejadores proporcionan un balance entre dos objetivos: la concurrencia y la integridad de los datos. Asegurar la integridad implica el bloquear los registros y/o archivos, para prevenir que dos o más usuarios actualicen el mismo dato al mismo tiempo. Para maximizar la concurrencia es necesario mantener al mínimo el nivel de bloqueos.

Una aplicación multiusuario que lleva a cabo operaciones de bloqueo de registros y archivos podría causar un problema de estancamiento (deadlock).

Esto ocurre cuando un proceso A bloquea un registro o archivo y necesita un registro o archivo que ha sido bloqueado por el proceso B y el proceso B necesita el registro o archivo que fue bloqueado por el proceso A. Cuando esto pasa los procesos esperarán indefinidamente a menos que el estancamiento sea detectado y resuelto por el sistema.





### Computadoras Personales

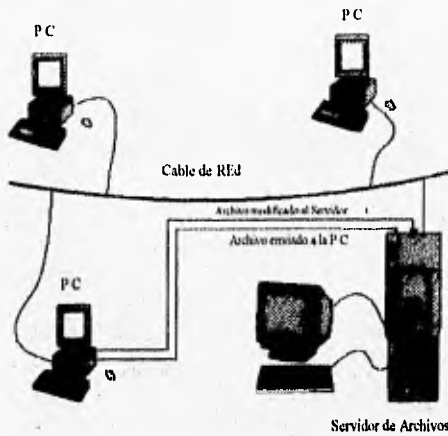
El manejador y las aplicaciones están combinadas formando un solo sistema. Los desarrollos con bases de datos en PC's contienen las pantallas de entrada, salida y consulta de los datos. El combinar estas tres funciones en una sola proporciona mayor poder, flexibilidad y velocidad al manejador, además de que el costo disminuye.

En una red de área local los datos y las aplicaciones se encuentran **generalmente** en el servidor, el cual, contienen además el sistema operativo de red. El servidor de la red **permite** que se puedan compartir datos entre los usuarios y diversos recursos como son las impresoras.

Todos los procesos del manejador son realizados en la PC. El servidor sólo busca en disco los datos requeridos por el usuario, y los envía a través de la red a la PC del usuario. Para cualquier cambio que se realice a la información es necesario que la PC envíe de regreso al servidor todo el archivo de datos para que sea nuevamente **almacenado** en el disco. Cuando varios usuarios están accedendo la base de datos, el servidor debe **enviar** a cada PC los datos que está solicitando lo que ocasiona un gran tráfico en la red disminuyendo así su velocidad.

Un problema de un sistema multiusuario es el de poder efectuar cambios **simultáneos** sobre un dato por diversos usuarios. Esto se soluciona **usualmente** agregando algún esquema de bloqueo de registros, en el que un registro o archivo es bloqueado cuando un **usuario** está modificándolo, impidiendo que otros usuarios puedan modificarlo al mismo tiempo.

Otro problema es el de realizar muchas transacciones **simultáneas** que incrementan el tráfico en la red, además de la limitante del poder de procesamiento de las PC's que **afecta** el **desempeño** del sistema. La solución que se ha desarrollado para estas limitantes es el uso de arquitecturas cliente/servidor.



Fuente: [SAL93]

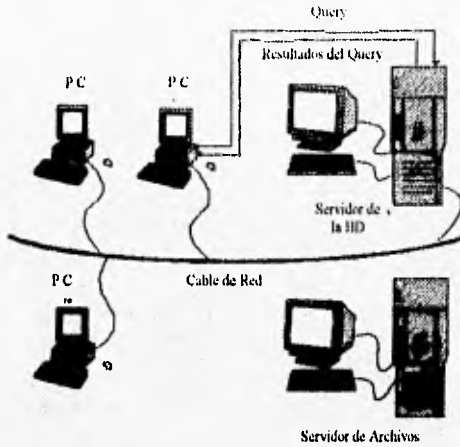
### Cliente/Servidor

En esta arquitectura el procesamiento de la base de datos se encuentra repartido en el cliente que generalmente son PCs y en el servidor. Los clientes se encargan de ejecutar la aplicación, y el servidor del manejador de base de datos y de compartir recursos como el espacio en disco, impresoras, etc.

Los clientes generalmente son PCs y el servidor puede ser desde una PC hasta un mainframe.

La aplicación que corre en los clientes es conocida como procesamiento frontal o interface de usuario; contiene todas las pantallas y procesos de entrada y salida de datos para el usuario. El servidor contiene el back-end, el cual se encarga del procesamiento de los datos y de los accesos a disco.

Una de las ventajas al compartir el procesamiento entre clientes y servidores es que se reduce el tráfico de datos en la red.



Fuente: [SAL93]

### Procesamiento Distribuido

Una base de datos distribuida, es aquella que no se encuentra almacenada en su totalidad en un solo lugar físico, sino que se distribuye a lo largo de una red de computadoras geográficamente separados que se conectan por medio de enlaces de comunicación. [DAT86]

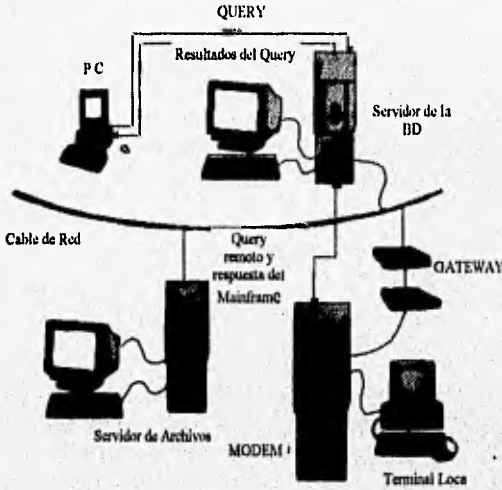
En esta arquitectura los datos son compartidos entre varios proveedores a través del envío de actualizaciones, así como de conexiones directas o remotas por medio de una línea telefónica o líneas dedicadas. Una aplicación que corre en uno o más procesadores extrae la porción de datos que va a ser modificada durante el proceso. Posteriormente se transmiten los datos a un host centralizado o a otros hosts que se encuentran distribuidos en el circuito. Las otras bases de datos son actualizadas también de tal manera que todo el sistema se encuentra en sincronía.

En un sistema distribuido, un usuario solicita información al host local, si los datos no se encuentran en el procesador, éste los busca a través de la red en todo el sistema hasta encontrarlos. Este proceso es transparente al usuario.

El usuario percibe a un sistema distribuido como si fuera un sistema centralizado ya que no necesita saber en dónde se encuentran físicamente almacenados los datos, es decir, que es completamente transparente para él. Así pues, el hecho de que la base de datos esté distribuida debe ser importante tan sólo a nivel interno, y no a niveles externos.

Las ventajas de tal distribución son que combina la eficiencia del procesamiento local (sin excesivos costos de comunicación) de la mayoría de las operaciones, confiabilidad y autonomía local.

Los problemas que se presentan en un sistema distribuido es el control global de concurrencia, actualización de datos duplicados, transparencia de ubicación y optimización de los procesos de consulta.



Fuente: [SAL93]

#### 1.4 SELECCIÓN DE LA ARQUITECTURA

Primero, se tiene que decidir si actualmente se necesita cambiar de arquitectura de base de datos. Un primer paso que hay que tomar en cuenta es, si múltiples usuarios accesan a la base de datos, si no es así, no hay por que cambiar a una arquitectura de tipo multiusuario y solamente con un sistema de computadora personal.

Si por el contrario se necesita acceso multiusuario, el siguiente cuestionamiento es cuantos usuarios pueden accesar simultáneamente, si la respuesta es menor a 20, ahora se debe preguntar si se cuenta con una red LAN. Si es afirmativa la respuesta un DBMS basado en PC multiusuario es más que suficiente. Si no se cuenta con una LAN se debe considerar uno de los menores sistemas operativos (así como uno basado en un multiusuario DOS, multiusuario OS/2, o UNIX), corriendo en una poderosa PC o en una minicomputadora RISC workstation. Las PCs existentes pueden actuar como terminales de el host multiusuario y una expansión futura puede ser por medio de terminales "tontas", dependiendo de las necesidades de los usuarios finales.

Si se cuenta con más de 20 usuarios simultáneos se puede determinar el tipo de acceso que los usuarios podrán tener a la base de datos. Si el acceso es para buscar un volumen de datos, y únicamente pocos usuarios pueden estar modificándolos (insertar, borrar y modificar), un DBMS multiusuario tradicional provee el desempeño aceptable para un número de usuarios entre 32 y 40. Sin embargo, si muchos o todos los usuarios de un DBMS tienen derecho a modificación de datos, se necesita de un sistema multiusuario poderoso y de la integridad de datos de una base de datos Cliente/servidor.

Otro aspecto que se debe tomar en cuenta es la flexibilidad o compatibilidad que debe tener la arquitectura elegida con el DBMS existente. Uno de los beneficios que nos ofrece la arquitectura cliente/servidor es que si se quiere hacer una sobrescritura completa, se pueden mover los datos a un servidor de archivos y usando una clase 4 de DBMS para proveer el acceso de los usuarios a los datos. Este método es el más fácil como una vía para mover un sistema existente a una arquitectura cliente/servidor.

## 1.5 ARQUITECTURA CLIENTE/SERVIDOR

### 1.5.1 Fundamentos

Una arquitectura que de unos años a la fecha se ha ido imponiendo en la arena del cómputo es la cliente/servidor (c/s). Si, durante mucho tiempo fue de los temas que, de forma casi obligada se hablaba, pero que en realidad poco se sabía. Ahora es de los más ampliamente tratados y para el que todos los fabricantes importantes de software, hardware y netware, ofrecen productos.

Una arquitectura es un conjunto de definiciones, reglas y términos, que se emplean para construir un producto. En consecuencia, la arquitectura cliente/servidor se apega a estos principios. Cliente/servidor fundamentalmente es un tipo de cómputo distribuido y cómputo cooperativo. Los actores, clientes y servidores, establecen una comunicación entre pares (peer-to-peer), o sea, de igual a igual dado que los clientes de una aplicación pueden actuar como servidores de otra.

Los equipos multiusuarios empezaron como un anfitrión con terminales tontas conectadas a ellos. En esta configuración, las terminales tontas actúan sólo como estaciones de entrada/salida y sin ninguna capacidad de cómputo asociada, dejando al anfitrión toda la carga de proceso y control. Los equipos maestro/esclavo constituyeron una etapa posterior en que la computadora maestra coordinaba el trabajo de los equipos esclavos, los que a su vez se encargaban de los equipos periféricos. La ventaja principal de esta configuración es que descarga el trabajo de entrada/salida a los equipos esclavos, pudiéndose entonces dedicar el equipo maestro al procesamiento propiamente dicho. La tercer etapa en los ambientes de cómputo es la cliente/servidor, en ella el cómputo se divide entre el servidor y el cliente, aprovechando mejor la potencia de cómputo de los equipos existentes hoy en día.

Hay un caso que seguramente habrá llamado su atención, el que una microcomputadora potente se utilice en emulación de terminal de una computadora mayor, lo que no exige gran parte de los recursos que puede ofrecer la microcomputadora. Por tal razón resulta, no sólo natural, sino deseable el que se aproveche la capacidad de cómputo de la microcomputadora, esto se lleva a cabo cuando las aplicaciones han sido desarrolladas para trabajar bajo la arquitectura cliente/servidor.

Las partes que intervienen en esta arquitectura son: el cliente, el servidor y el elemento que los enlaza a la red de cómputo. Tanto el cliente como el servidor son combinaciones de software y hardware. Así, por ejemplo, una computadora puede actuar como cliente sólo si también tiene instalado el software cliente; la máquina en sí misma no tiene conferida la capacidad del cliente. En un ambiente cliente/servidor operan al menos dos procesos: el cliente que inicia la comunicación y el servidor que, estando a la espera de requerimientos, la responde.

También son tres los elementos que se distribuyen entre el cliente y el servidor: el manejo de datos, la aplicación y la presentación. El manejo de datos se refiere al sistema de archivos o al manejador de bases de datos, la aplicación es el software que emplea tales datos para el propósito específico del usuario y la presentación es el software que establece la forma en que los datos se visualizarán en la pantalla del cliente. [BYTE]

Existen básicamente dos grupos definidos en esta tecnología, el servidor de archivos (usualmente bases de datos) y el servidor de aplicaciones (programas).

El servidor de archivos requiere de un manejo de entrada y salida de datos muy eficiente, tanto en la lectura de datos de los sistemas de almacenamiento (discos/cintas), como de los sistemas de comunicación (cableados).

Por otro lado, el servidor de aplicaciones requiere de un procesador muy veloz, para dar respuesta oportuna a las solicitudes de proceso que reciba. Estos conceptos no son nuevos, existían con las computadoras grandes y las minicomputadoras.

Todos los equipos de cómputo son más valiosos por los datos que contienen que por su valor en sí, cualquier falla significa pérdidas, pero hay fallas que pueden esperar a ser reparadas y otras que tienen que ser reparadas de inmediato, a estas aplicaciones se les ha dado en llamar de "misión crítica", un ejemplo, son las instalaciones de las sucursales de los bancos, donde los clientes no pueden esperar a que regrese el sistema.

Una de las grandes diferencias son las instalaciones de mainframes y terminales, es que en éstas todo el proceso de almacenamiento descansa en una sola computadora, en el concepto actual, tanto el equipo cliente/servidor, como las microcomputadoras conectadas a él, incluyen sistemas de almacenamiento y microprocesadores independientes que procesan datos aisladamente y que pueden ser colocados en el cliente/servidor para que otros usuarios conectados a él los puedan consultar o modificar.

En pocas palabras, podemos decir que se trata de una red de microcomputadoras conectadas a otra microcomputadora central que sirve para almacenar información común, tanto de datos como de aplicaciones y que puede o no, estar conectada a otras redes o macrocomputadoras [PC MONITOR]

Para la perspectiva de los desarrolladores, una de las más importantes diferencias entre las aplicaciones típicas propietarias multiusuario y aplicaciones no propietarias cliente/servidor es el mayor conocimiento de los fundamentos de las tecnologías requeridas.

Desarrollando aplicaciones cliente/servidor basadas en tecnologías no propietarias cliente/servidor, se requiere de un completo conocimiento del medio ambiente en conjunto. La aplicación misma es ahora separada físicamente de la base de datos por capas de componentes sustituidos por diferentes vendedores que a menudo interactúan en nuevos y novedosos caminos.

Adicionalmente, la aplicación misma es localizada físicamente en la plataforma del usuario, el medio ambiente de hardware y software es esencialmente controlado por el usuario y no por un equipo de alto apoyo técnico supervisado por comités de control de cambio y las otras infraestructuras organizacionales que se desenvuelven en el conocimiento de las necesidades de grandes centros de datos típicos. El desarrollador puede ya no asumir que cualquier cosa va a trabajar como se especifica y cuando aparezcan problemas, por más inevitables que sean, el desarrollador puede ser bastante conocedor para identificar cuando estos problemas ocurren.

Para ser acertado en un ambiente cliente/servidor, las aplicaciones desarrolladas deben ser más generales que la típica contraparte de mainframes y puedan invertir más tiempo en adquisición de nuevas habilidades y conocimientos. Los fomentadores de Cliente/servidor no pueden depender de expertos en un simple lenguaje, transacciones de monitor y sistemas manejadores de bases de datos para verlos durante 20 años y retirarse. La complejidad es mucho mayor y el grado de cambio es muy rápido.

Las nuevas noticias son que los desarrolladores de Cliente/servidor no deben ser expertos en todas las facetas de su medio ambiente, sólo suficientemente conocedores para entender las interacciones entre los componentes. En muchos equipos de desarrollo de proyectos Cliente/servidor existen uniones de profundos expertos técnicos que planean el levantamiento y mantenimiento del lugar de trabajo, así también, solucionan problemas en la localización de averías que se presentan. Pero el desarrollador, como la persona directamente responsable para la derivación de la aplicación, debe ser hábil para reconocer los síntomas del problema y después tomar acciones correctas independientes o llamar al conjunto apropiado de expertos. Adicionalmente, es el desarrollador quien ha de cerrar contactos con el usuario final, y nunca olvidar, que es el usuario final quien ejerce el control actual bajo el lugar de trabajo del "cliente".

### Características de una Arquitectura Cliente/Servidor

En el modelo cliente/servidor podemos encontrar las siguientes características:

1. El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas realizando actividades o tareas independientes.
2. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
3. Un servidor da servicio a múltiples clientes concurrentemente.
4. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los clientes o de los servidores ya sean por actualización o por reemplazo tecnológico, deben realizarse de una manera transparente para el usuario final.
5. La interrelación entre el hardware y el software debe de estar basada dentro de una infraestructura poderosa de tal forma que el acceso de los recursos de red no muestre la complejidad de los diferentes tipos de formato de datos y de los protocolos.

Un sistema de servidores realiza múltiples funciones, al mismo tiempo que presenta una imagen de un sólo sistema a las estaciones clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un sólo sistema lógico proporcionando de ésta manera el servicio más efectivo para el usuario final. También es importante hacer notar que las funciones cliente/servidor pueden ser dinámicas, ejemplo: un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.

En conclusión, cliente/servidor puede incluir múltiples formas, bases de datos, redes y sistemas operativos, estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias funcionando todos al mismo tiempo. por lo tanto, su implantación involucra diferentes tipos de estándares; APPC, TCP/IP, OSI, NFS, DRDA corriendo DOS, OS/2, Windows o Unix, en Token-Ring, Ethernet, FDDI o medio coaxial, solo por mencionar algunas de las posibilidades.

Dependiendo de estas características y su combinación, resulta el grado de complejidad de una solución cliente/servidor. [RED]

La característica más importante es la confiabilidad, el cliente/servidor no puede fallar, no puede perder la información ahí guardada, pues de ello depende la operación de la empresa. Este es el aspecto que debe determinar la elección de un equipo u otro, que los sistemas de almacenamiento y seguridad sean de excelente calidad y estén totalmente garantizados por el fabricante.



Dentro de los sistemas de seguridad que los cliente/servidor deben incluir, está el informar de alguna manera el tipo de falla que han tenido, el reiniciar las operaciones sin pérdida de información, el contar con un procesador de respaldo, bien sea en estado de espera por si falla el procesador central o funcionando en paralelo, pero que tenga la habilidad de hacerse cargo de las funciones que ejecutaba el procesador que ha fallado.

Antes de decidir por la marca de un equipo que ofrezca las bondades de esta tecnología, es importante decidir si esta tecnología es la conveniente a la empresa. Para muchas empresas una red con la que pueden compartir periféricos y ocasionalmente archivos entre usuarios es suficiente. Pero si la información de su empresa debe ser consultada y/o modificada por diferentes personas, además de compartir algunos otros recursos, como son los periféricos, entonces conviene planear la adquisición de un equipo confiable para que haga las funciones de cliente/servidor. [PC MONITOR]

### Consideraciones para la Implantación de Cliente/Servidor

El modelo cliente/servidor es muy sencillo en su conceptualización, pero su implantación no necesariamente lo es. en el momento en que una compañía empieza a distribuir sus aplicaciones y servicios de cómputo, la administración de sistemas y la definición de donde ubicar los datos deben ser seriamente considerados y evaluados. Un solo sistema de LAN puede requerir poca administración o nada de ella, sin embargo en un ambiente de cómputo cliente/servidor empresarial, las facilidades de administración de sistemas son requisito indispensable para una implantación exitosa.

Además, de lo anterior existen otros puntos a considerar:

1. **Reforzar la seguridad de acceso.** En los sistemas distribuidos, el acceso es frecuentemente limitado a proteger ciertos datos de un servidor dado. Sin embargo, no hay ningún control físico, lógico o administrativo para el control de datos usados por múltiples PC's, excepto para controlar el medio físico de almacenaje (discos, cintas, etc.)

2. **Mantener la integridad.** Es necesario definir quien es el responsable de mantener la integridad de los datos y de las aplicaciones que se encuentran distribuidos en las redes que componen el sistema, estableciendo políticas y procedimientos de seguridad.

3. **Soporte de proveedores.** Otro problema potencial es el soporte y el servicio. Una red es generalmente integrada con productos de varios proveedores. Entre mas proveedores involucrados en un sistema, más difícil es la determinación de los problemas. esta situación llega a empeorarse por la falta de experiencia en ambientes distribuidos y heterogéneos.

### **¿Cuándo se requiere de un Cliente/Servidor?**

Para muchas empresas, una red para compartir algunos recursos es suficiente. Las empresas que necesitan compartir impresoras, faxes, scanners, cualquier periférico o eventualmente la información de otros usuarios, no requieren de la tecnología cliente/servidor, basta con tener una red bien instalada.

Las empresas que tienen grupos de usuarios con la necesidad de compartir la misma información, tanto para consulta como para ser modificada, pueden sacar provecho de esta tecnología. Generalmente la información compartida son datos en forma de bases de datos. Las bases de datos pueden contener información de productos, personas, publicaciones y mucho más.

Una empresa que tiene constantes movimientos de mercancía requerirá que los datos correspondientes a los productos que vende o desplaza estén a la mano de las personas que hacen pedidos o realizan ventas, para saber si requieren hacer pedidos a sus proveedores o tienen los productos para entrega inmediata que sus clientes solicitan, por otro lado, el departamento de contabilidad empleará los datos para facturación, cuentas por cobrar o pagar. [PC MONITOR]

### **Desarrollo de Aplicaciones Cliente/Servidor**

Para el desarrollo de aplicaciones cliente/servidor se puede emplear un rango de herramientas que van desde lenguajes de tercera generación, hasta ambientes de desarrollo, no siendo exclusivo de algún tipo de herramienta, el desarrollo de aplicaciones cliente/servidor.

**Lenguajes de tercera generación.** Son los lenguajes tradicionales con extensiones para su interfaz con la red de cómputo y con las bases de datos remotas. Ejemplos de este caso son: los lenguajes C y COBOL, con las bibliotecas de APIs adecuadas (por sus siglas en inglés: Application Program Interface, Interfaz para Programa de Aplicación).

**Lenguajes de cuarta generación.** Muchos de los fabricantes de manejadores de bases de datos importantes, como Informix, Oracle, Progress y Sybase, ofrecen herramientas de desarrollo consistentes en lenguajes de cuarta generación. Estas pueden emplearse en desarrollos cliente/servidor para sus bases de datos. Además, muchos otros ofrecen lenguajes de cuarta generación que pueden operar con una variedad de bases de datos con los controladores correspondientes.

**Ambientes de desarrollo.** Un ambiente de desarrollo es mucho más que un lenguaje de cuarta generación, es un entorno de trabajo completo. Además de incluir un 4GL y su editor, también incluye un proveedor (generalmente basados en objetos) de procedimientos, mensajes de error, mensajes de ayuda, atributos de las entidades, etc. Además incluye herramientas de control de versiones, herramientas de diseño, herramientas de desarrollo de

prototipos, herramientas de usuario final y utilerías de referencias cruzadas, por mencionar tan solo los recursos ofrecidos con mayor frecuencia.

Por otro lado, muchas herramientas CASE tienen compuertas con lenguajes de tercera y cuarta generación, además, con ambientes de desarrollo. Esto permite la comunicación de y hacia estas herramientas.

### **Otras Aplicaciones de la Arquitectura Cliente/Servidor**

Además de las aplicaciones de bases de datos con clientes gráficos, la arquitectura cliente/servidor se manifiesta en muchas otras aplicaciones, por ejemplo, en equipos Unix. Otro tipo de sistema operativo de red que ahora están ofreciendo es el NFS ( por sus siglas en inglés: Network File System, Sistema de Archivos en Red). Bajo este sistema el disco duro de los servidores puede ser empleado como disco de los clientes, permitiendo guardar en ellos archivos de datos, programas y en general, cualquier cosa que se pudiera archivar en los discos locales de los clientes.

Otro caso de aplicaciones cliente/servidor son los servicios de Internet como Gopher y World Wide Web. En los servidores residen los archivos y en los clientes se instala el software de consulta y visualización.

Aún cuando la tecnología cliente/servidor no es nueva, sí es relativamente reciente.

Podríamos argumentar que sus beneficios son grandes, pero la mejor manera de comprobarlo es utilizándola en una misma aplicación, desarrollada en dos versiones. Una de éstas en forma convencional y otra en cliente/servidor, la diferencia es evidente. No es pues de extrañar, que muchas de las aplicaciones que tradicionalmente emplean el servidor de la red como servidor de archivos estén siendo reconstruidas para operar en otras modalidades de cliente/servidor en que se aproveche la capacidad de cómputo del servidor, y no únicamente su capacidad de almacenamiento y de compartición de archivos.

### 1.5.2 Ventajas y Desventajas

Las ventajas más importantes son:

- Permite un mejor aprovechamiento de la potencia de cómputo de los equipos al poder descargar en el cliente parte de la carga de trabajo del servidor.
- Permite el acceso de un cliente a varios servidores en forma simultánea, lo que resulta cada vez más importante en los ambientes heterogéneos actuales.
- Reduce el tráfico en la red, ya que por ella sólo viajan los requerimientos y las atenciones a ellos.
- La arquitectura cliente/servidor es particularmente adecuada para los sistemas de Procesamiento de Transacciones en Línea ya que cumple sus cinco postulados básicos.
  1. Los datos de las transacciones se ingresan a la computadora
  2. Se trae a la memoria el registro maestro correspondiente
  3. Se actualiza el registro maestro en la memoria
  4. Se actualiza el registro maestro en disco
  5. El registro de transacciones se agrega al archivo de transacciones
- Puede y, de hecho, tiene que operar bajo sistemas abiertos, lo que significa capacidad de cambio de plataformas con un mínimo de problemas y riesgos.
- Permite el uso de interfaces gráficas de usuario muy versátiles y amigables en los clientes

Como desventajas tenemos que las aplicaciones cliente/servidor suelen ser más complejas que las tradicionales en la forma de anfitrión/terminales, y que exigen más de la red. En ambos casos, las desventajas pueden ser superadas mediante las ventajas anotadas, y ciertamente, todo tiene su costo.

Los expertos en seguridad informática mantienen, que bajo cliente/servidor la seguridad es más sencilla de romper, pero la respuesta a esta aseveración es que la seguridad está sujeta al diseño y medidas de seguridad que se incorporen al ambiente, y a las aplicaciones desarrolladas para operar bajo la arquitectura cliente/servidor

### 1.5.3 Seguridad

En los sistemas de información, la seguridad es un punto que no puede pasarse por alto. Consiste en una serie de medidas que protegen a la base de datos contra accesos no autorizados, alteraciones erróneas intencionales y pérdida de la integridad referencial de los datos. El administrador de la base de datos es el responsable de administrar y monitorear la seguridad de los datos, asignación de claves de acceso (passwords con sus respectivas restricciones), y verificar violaciones de seguridad. El nivel de seguridad que se pueda alcanzar dependerá fuertemente del sistema operativo y del software de base de datos que se utilice, cabe mencionar que muchos de los DBMS para PC's ofrecen muy poca seguridad y no conservan historial (logs) de las actividades de los usuarios, a diferencia de los DBMS creados para los servidores de bases de datos que, como se indicó anteriormente, contribuyen a la seguridad de los datos, previniendo que el usuario accese a los archivos desde afuera del DBMS, ya que estos se encuentran encriptados y si guardan un registro de las transacciones que realiza el usuario.

Otras medidas de seguridad de los DBMS es que cuando se va a realizar un conjunto de transacciones (instrucciones) que modifique a más de una tabla, estas pueden tratarse como si fuera una sola instrucción a través de herramientas conocidas como commits, cuyo objetivo es garantizar que se ejecuten todas o ninguna de esas instrucciones, protegiendo así la integridad de la información. Los passwords son usados para controlar el acceso a la aplicación entera y, dependiendo del DBMS, también para el uso de menús, formas, reportes, tablas y hasta determinadas columnas de dichas tablas.

Al trabajar en red, diversos usuarios pueden tener acceso a las tablas que integran a la base de datos, esto comúnmente origina el problema de que simultáneamente se quiere modificar una tabla por dos usuarios. Este problema se resuelve bloqueando. El bloqueo presenta un problema más, llamado dead lock pero es de mencionar que hay DBMS's que resuelven este problema en forma automática porque detectan este conflicto y le dan prioridad al usuario que primero accedió. Un enfoque diferente de visualizar la seguridad es el que la divide en las siguientes características:

- **Física.** El lugar que contienen los sistemas de cómputo debe ser físicamente seguro contra la entrada de intrusos.
- **Humana.** la autorización de usuarios debe ser hecha de manera cuidadosa para reducir el riesgo de que un usuario autorizado por medio de un soborno o como un favor, le de acceso a una persona no autorizada.
- **Sistema Operativo.** No importa que tan seguro sea el DBMS, si el sistema operativo es débil en cuanto a seguridad puede permitir accesos no autorizados a la base de datos.
- **Sistema de base de datos.** Desde la misma base de datos se puede ejercer un control muy importante que tiene que ver con la asignación adecuada de permisos. De este

modo, a algunos usuarios se les podría autorizar tener acceso a toda la información, a otros se les podría permitir realizar consultas a una porción limitada de la base de datos y prohibirles realizar modificaciones. Será responsabilidad del DBMS asegurar que esas restricciones no sean violadas. [KOR86]

### **Aplicaciones de Seguridad a Bases de Datos Cliente/Servidor**

“La buena noticia acerca de las bases de datos relacionales es que permiten el acceso de los usuarios a los datos, pero la mala es que permite el acceso de los usuarios a los datos.”

Conforme se extiende el uso de la tecnología cliente/servidor se hace más común la creación de aplicaciones dedicadas para acceso remoto. Entre los primeros usuarios se tiene a los bancos e instituciones financieras en general, ya que poseen grandes cantidades de información en sistemas centralizados y cada día hay más usuarios remotos que exigen el acceso a dicha información. Según Bobby Cameron (analista senior de Forrester Research): “La buena noticia acerca de las bases de datos relacionales es que permiten el acceso de los usuarios a los datos, pero la mala es que permite el acceso de los usuarios a los datos.”

Los principales retos a los que se enfrenta un sistema de bases de datos relacionales (RDBMS) se refieren a la protección de los datos en el servidor y a la restricción de la información que sale y entra desde sitios remotos.

### **Protección de Datos en el Servidor**

Una buena parte del problema de seguridad reside en el hecho de que, tradicionalmente, el acceso es controlado a través del software administrador de la base de datos en lugar de ser controlado por la propia base, lo que hace difícil de implantar esquemas de seguridad. Este problema está siendo atacado por algunos de los proveedores de la industria que ofrecen productos que integran mecanismos de seguridad en la base de datos misma, lo cual ayuda a resolver la gran cantidad de riesgos que existen hoy en día.

Un ejemplo muy claro de lo anterior lo constituye el Maryland National Bank que eligió un producto en particular debido al uso de procedimientos almacenados (stored procedures), los cuales incrementan la habilidad de los programadores para establecer una seguridad adecuada. Un procedimiento almacenado es un tipo especial de subrutina que está dentro del servidor de la base de datos y que actúa como filtro o control entre el usuario y la información que llega o sale de la misma. Aquí el truco estriba en lograr que tanto la base de datos (el engine) como el sistema que la administra, proporcionen herramientas que permitan mantener la seguridad de la información. Así pues, el Maryland National Bank ha establecido un esquema con tres niveles de seguridad para los sistemas cliente/servidor:

- 1 - Seguridad a nivel de red mediante restricciones en el login con horarios de acceso limitados para firmarse en la red, acceso restringido a ciertos recursos y contraseñas con fechas de caducidad.
- 2 - Seguridad a nivel de las aplicaciones desarrolladas por la empresa.
- 3 - Seguridad interconstruida en el sistema administrador de base de datos relacional, RDBMS, a nivel de procedimientos almacenados.

#### **Restricción de accesos remotos**

Por otro lado, la protección para accesos remotos no autorizados al sistema, sobre todo de piratas informáticos, ha ido en aumento conforme son más las compañías que integran este tipo de servicios tanto por canales privados como públicos (líneas telefónicas, por ejemplo). Para ello existe una gran cantidad de productos en el mercado que contemplan diversos métodos de protección, entre los que se puede mencionar los siguientes:

- 1.- **Llamada de regreso o retrolamada (call-back).** El usuario llama al host (servidor o sistema central), se identifica y corta la comunicación. Si la llamada se realizó desde un número telefónico autorizado el sistema valida al usuario y regresa la llamada a dicho número para permitirle el acceso.
- 2.- **Verificación por localizador.** El usuario se conecta al sistema y se identifica con el mismo, entonces el administrador le envía un número de acceso o contraseña a su localizador (pager) para que pueda entrar al sistema. Este número es aleatorio.
- 3.- **Utilización de un programa residente en memoria (TSR, terminate and stay resident) en el sistema remoto.** Este software se encarga de la verificación y autenticación ante el sistema central y es la opción que más aceptación tienen entre los usuarios puesto que los libera de la mayor parte del trabajo realizado para validarse y entrar al sistema.
- 4.- **Sistema de autenticación por señal (token).** En este método se usa un aparato similar a una calculadora que está sincronizado con el sistema central. Cuando el usuario se conecta, el servidor le envía un número que debe alimentar al aparato, el cual, de acuerdo al número alimentado, emitirá una respuesta única para ser usada como contraseña por el usuario y, debido a que el algoritmo para la generación de la contraseña es el mismo en ambos lados de la conexión, solo se permitirá el acceso cuando se tenga la señal correcta.

### El Futuro

No importa la cantidad de medidas involucradas ni que tanto haya costado establecer un sistema de protección de datos si no se toma en cuenta la retroalimentación de los usuarios para buscar y corregir fallas que puedan causar errores en la información almacenada.

El continuo depuramiento es una de las herramientas más útiles para mantener una base de datos a prueba de intrusos; sin embargo hay que tener mucho cuidado con la sobreingeniería de las aplicaciones para evitar el gasto innecesario de recursos en la empresa. Debe encontrarse el punto de equilibrio en el que se tiene un sistema razonablemente seguro sin tener que invertir mucho tiempo en su desarrollo y depuración. Si lo anterior no se toma en cuenta puede llegarse a casos extremos en los que después de muchos meses de trabajo se libera una aplicación "totalmente segura" que resulta obsoleta pues los requerimientos de los usuarios han cambiado con el tiempo.

Otro aspecto importante por resolver es la falta de herramientas para el control centralizado de las diversas plataformas involucradas en un sistema cliente/servidor (mainframes, redes locales, redes de área amplia, etc.). Hasta ahora los administradores deben emplear herramientas propias de cada plataforma para los diversos componentes del sistema. Este problema se irá resolviendo conforme surjan nuevos estándares de conectividad entre plataformas.

Finalmente, y esto es muy importante, es necesario educar a los usuarios para que sigan las políticas de seguridad establecidas. De todos es conocida la historia de compañías en las que es muy fácil entrar a sus sistemas centrales porque los usuarios tienen la contraseña pegada en su monitor, en el teclado, etc. y cualquier persona podría hacer mal uso de ella. [RED]





**CAPÍTULO II**  
***COMPONENTES DE LAS***  
***APLICACIONES***  
***CLIENTE/SERVIDOR***



## **CAPÍTULO II**

### **COMPONENTES DE LAS APLICACIONES CLIENTE/SERVIDOR**

#### **OBJETIVO DEL CAPÍTULO**

Explicar claramente como es que se integran las aplicaciones de la arquitectura cliente/servidor y especificar claramente cada uno de sus componentes.

#### **INTRODUCCIÓN**

La arquitectura cliente/servidor, es una tecnología de punta, pero que, por aspectos comerciales, el termino ha caído en una especie de moda, donde todas las aplicaciones según sus fabricantes son cliente/servidor.

Primero se establecerá de manera sencilla el concepto. Se debe de partir que en una red de área local (LAN, por sus siglas en inglés) existe un proceso distribuido, donde el procesador del Servidor ejecuta las instrucciones del Sistema Operativo de RED, generalmente de servicios de archivos; y el procesador de las Estaciones de Trabajo procesan los trabajos locales, pero las aplicaciones cada día requieren de mayor poder en el procesador, lo que implicaría que las Estaciones de Trabajo tuvieran procesadores muy poderosos, con las respectivas consecuencias económicas, pero si se parte del echo que los Servidores cuentan con este tipo de procesadores, la arquitectura cliente/servidor, implica aprovechar estos procesadores poderosos para hacer las tareas pesadas y los trabajos ligeros dejárselos al procesador de las Estaciones de Trabajo.

La ventaja del modelo cliente/servidor, es que se permite que las tareas se repartan en forma más eficiente entre los elementos involucrados (clientes-servidores) y que se minimice el intercambio innecesario de información entre ellos.

En la actualidad la arquitectura cliente/servidor, se explota principalmente en las aplicaciones de bases de datos, y se partirá de este ejemplo para ilustrar sus componentes.

El primer elemento es el motor de bases de datos (Database engine), tales como el SQL Base de Gupta, el NetWare SQL de Novell, el SQL Server de Microsoft, el Oracle SQL, o el SYBASE SQL Server 10 de Sybase, por mencionar algunos. Este tipo de productos NO SON un manejador de base de datos, sino es la maquinaria de base de datos que da la plataforma para que una computadora se convierta en un servidor de base de datos.

El segundo elemento es un manejador de base de datos que corre en dos computadoras al mismo tiempo, en el cliente y en el servidor. La parte que corre en el servidor de base de datos se denomina back-end y la que corre en el cliente se llama Interfaz

de Usuario o front-end. Algunos ejemplos de esto son Dbase, Clipper, Revelation, Informix, Paradox, XQL, SQL Windows y el mismo SYBASE SQL Server 10 de Sybase.

En las aplicaciones tradicionales de base de datos, es el archivo completo el que viaja desde el servidor de archivos a la estación de trabajo. Esta última procesa la información después de una serie de 'preguntas y respuestas' que se llevan a cabo entre el servidor y la estación y finalmente devuelve el archivo. En el modelo cliente/servidor, la estación de trabajo (el cliente) pregunta al servidor de base de datos a través del front-end y del back-end. El servidor procesa la información y devuelve el resultado, el registro, no el archivo completo que el cliente demandó.

El proceso cliente/servidor que se lleva a cabo en una gateway de comunicaciones es muy similar. La computadora gateway (servidor) que se encarga de las labores de comunicación entre el procesador central (host) y la estación de trabajo (cliente), le envía a este último solamente las pantallas necesarias y toma la información del teclado, pero el proceso de comunicaciones se lleva a cabo por completo en el gateway de comunicaciones.

Algunas otras aplicaciones más sencillas como servidores de comunicaciones, de faxes o de impresión, comienzan a explotar este modelo.

Sin duda, el modelo cliente/servidor es el que le da a las redes locales la posibilidad de ir más allá de servicios de impresión o de archivos y el que permite que el proceso de información sea verdaderamente distribuido. Continuamente surgen aplicaciones en este modelo para todos los sistemas operativos de red. Esto con seguridad ayudará a desarrollar el verdadero proceso distribuido, que convierte a las Redes LAN de computadoras en herramientas únicas del mundo de la computación.

## II.1 EL CLIENTE

Hemos considerado al cliente como una computadora, ya sea mini, PC o estación de trabajo conectada a una red que se utiliza para acceder información y recursos de la red, así como sus propios recursos. Los programas del lado del cliente representan una parte del total de la aplicación que le permite al usuario el interactuar con la misma. Esto típicamente es una interfaz que le permite iniciar la aplicación, solicitar datos y desplegar información. Para el usuario debe verse como si estuviera trabajando en una computadora stand-alone.

La necesidad de mayor poder en las herramientas desarrolladas para los clientes ha crecido tanto como el equipo ha evolucionado, el cliente fué típicamente una terminal o una PC 8088 con 256 Kilobytes de memoria y tal vez 10 megabytes de almacenamiento en disco. Por lo que las aplicaciones estaban capacitadas para realizar sólo algunos cálculos y usaban una significativa porción de su poder en presentación y en la lógica de interfaz de usuario. La memoria limitada también restringía la complejidad de cálculos que se podían ejecutar en esas máquinas. Como resultado, las interfaces de los clientes a los servicios eran diseñadas

para minimizar el uso de procesador central y la utilización de memoria en el cliente y maximizaba el uso del poder de procesamiento del servidor.

El equipo moderno del cliente ha establecido el uso de tecnologías avanzadas de presentación tales como Interfases Gráficas de Usuario, así como avanzadas tecnologías de desarrollo incluyendo tecnología orientada a objetos.

Las aplicaciones del cliente pueden usar gran capacidad de memoria para retener grandes volúmenes de datos complejos así como para poder ejecutar operaciones complejas con los mismos. Las interfaces del cliente deben tener la capacidad de mover los datos desde los servidores a la memoria disponible de éste eficientemente.

Durante los pasados diez años, el funcionamiento de la estación de trabajo se ha mejorado dramáticamente. Por el mismo costo, el funcionamiento del procesador de la estación de trabajo es aumentado 50 veces, la memoria principal ha aumentado 25 veces, y el disco duro ha aumentado 30 veces. Este crecimiento en potencia permite muchas aplicaciones más sofisticadas para ser corrido desde el escritorio.

Las comunicaciones y la Interfaz de Programa de Aplicación de las redes han mejorado igualmente en los últimos diez años. En 1984, el funcionamiento y la velocidad de un archivo remoto, una base de datos y servicios de impresión, fueron inadecuados para mantener las aplicaciones comerciales. Con la llegada de las redes locales de alta velocidad y áreas extensas (LAN y WANs, por sus siglas en inglés), protocolos de redes, conmutadores digitales y cableado óptico el funcionamiento y la confiabilidad mejoraron substancialmente. Esto es ahora práctico para usarse en esos servicios remotos como parte de una aplicación comercial.

La estación de trabajo del cliente puede usar DOS, Windows, Windows NT, OS/2, MacOS (también referido como un sistema 7), o sistema operativo UNIX. Cuando la estación de trabajo del cliente es conectada a una LAN, esta tiene un acceso a los servicios proporcionados por el sistema operativo de la red (NOS, por sus siglas en inglés) en adición los proporcionados localmente por la estación de trabajo. La estación de trabajo puede cargar el programa y salvar los documentos procesados desde un servidor y por lo tanto usa funciones del archivo del servidor suministradas a través del NOS. Las estaciones de trabajo del cliente pueden ser usadas como una terminal para acceder las aplicaciones residentes en un computador central (host) de una minicomputadora en un procesador de una mainframe. Esto permite que la estación de trabajo reemplace a la terminal permitiendo así una mayor funcionalidad.

En una aplicación cliente/servidor, las funciones son proporcionadas por una combinación de fuentes usando el procesador de la estación de trabajo (cliente) y el procesador del servidor. Por ejemplo, un servidor de base de datos suministra datos en respuesta a la petición expedida del SQL por la aplicación del cliente.

El procesamiento local para el cliente podría calcular la suma de la factura y formatear la respuesta en la pantalla de la estación de trabajo.

Las estaciones de trabajo pueden proporcionar funciones comerciales usando una mezcla de productividad personal que se realiza en conjunción con una aplicación de costumbre. Por ejemplo, un documento creado por un procesador de palabras puede incluir una entrada desde una hoja de cálculo los datos de la factura creados por la aplicación cliente/servidor. La capacidad para cortar y pegar una entrada desde diferentes fuentes es uno de los aspectos más poderosos de la estación de trabajo, ya que esta le proporciona al usuario final las herramientas para crear nuevas aplicaciones.

### II.1.1 El Papel del Cliente

En el modelo cliente/servidor, el cliente es primeramente un consumidor de servicios provistos por uno o más procesadores del servidor. El modelo proporciona una clara separación de funciones basadas en la idea de los servidores actuando como proveedores de servicio respondiendo a las peticiones de los clientes. Esto es importante para entender que una estación de trabajo puede operar como un cliente en algunas ocasiones mientras actúa como servidor en otras ocasiones. Por ejemplo, en un ambiente administrador LAN, una estación de trabajo podría actuar como un cliente para un usuario mientras actúa simultáneamente como un servidor de impresión para muchos usuarios.

La tecnología actual provee un soporte efectivo de costo para una Interfaz Gráfica del Usuario (GUI). El ambiente de Windows permite al usuario (cliente) estar involucrado en varias sesiones simultáneamente. Tales funciones como procesamiento de palabras, hoja de cálculo, correo electrónico y representación de gráficas - en suma con las aplicaciones de costumbre construidas por la organización - pueden ser activadas simultáneamente. Windows 3.X y Mac System 7 no soportan multitarea; de esta manera, el manejo de una tarea puede interrumpir las comunicaciones. Windows NT, OS/2 y UNIX, son sistemas operativos multitareas apropiados para mantener el manejo de sesiones de comunicación.

Las facilidades obtenidas tales, como el Intercambio Dinámico de Datos (DDE, por sus siglas en inglés), Enlace y Empotramiento de Objetos (OLE, por sus siglas en inglés), la Arquitectura Corredor Expedida del Objeto de Comunicación (CEOC), proporcionan un soporte para las operaciones de cortar y pegar entre los procesadores de palabra, las bases de datos, hojas de cálculo y gráficas en el ambiente de Windows. Así, un conjunto seleccionado de tareas puede ser ejecutado por el cliente. En realidad la estación de trabajo puede ser tanto cliente como servidor cuando toda la información y lógica pertinente a una petición es residente y opera dentro de la estación de trabajo del cliente.

El programa para soportar funciones específicas -por ejemplo, ediciones de campo, ayuda sensitiva de texto, navegación, entrenamiento, almacenamiento personal de datos, y la manipulación- frecuentemente se ejecuta en la estación de trabajo (cliente). Todas estas funciones usan la Interfaz Gráfica de Usuario y la funcionalidad de Windows.

Una cliente usa un sistema operativo local para el servidor de servicios básicos y las interfaces del sistema operativo de la red. Este sistema operativo puede ser el mismo o diferente que el que usa el servidor. La mayoría de los de los usuarios personales de computadoras ahora usan DOS o Windows 3.X como su sistema operativo para el cliente, porque los usuarios comunes usan aplicaciones de productividad personal.

Estos usuarios corren aplicaciones cliente/servidor desde DOS o Windows únicamente como un simple proceso comercial en un tiempo. Sin embargo, la demanda para usar esos sistemas operativos es manejar el funcionamiento de las herramientas del cliente/servidor tales como el Power Builder para Windows, y las nuevas versiones de multitarea de Windows, tales como Windows NT, Windows 95 que ya están disponibles. Afortunadamente, la llegada de productos como el Digital's Parts y el Parc Place's Visual Works proporcionan herramientas de desarrollo que igualmente corren en Windows 3.X o OS/2, UNIX, Windows 95 y Windows NT.

A pesar de que UNIX y OS/2 han carecido de las herramientas de productividad personal tales como procesadores de palabras, correo electrónico, hojas de cálculo, presentación de gráficos y sistemas administradores de base de datos, DOS y Windows XX se han convertido en sistemas operativos seleccionados para el cliente. Ahora los progresos en la capacidad de otros sistemas operativos para correr aplicaciones de productividad personal, y el incremento del usuario que necesita una alta confiabilidad o multitarea, han aumentado la popularidad de las terminales -X, OS/2 y UNIX. La ejecución de las aplicaciones nativas de Windows 3.1 bajo el Windows NT, OS/2 o muchas implementaciones de UNIX ofrecen mejoras para el usuario de escritorio: confiabilidad y funcionalidad.

La disponibilidad que existe del OS/2 versión 2.1, UNIX, Windows NT, Windows 95 con soporte integrado para DOS, Windows 3.X y X-Windows (también como un soporte para multitarea en un ambiente confiable), es una razón para hacer que los sistemas operativos del cliente usen la selección para desarrollar aplicaciones críticas comerciales del cliente/servidor. Como se sabe, la reducción dramática en los costos del procesador y los costos de la ampliación de RAM hacen que los recursos extras requeridos para OS/2, UNIX, Windows NT y Windows 95 sean mínimos. Finalmente los costos de la licencia de los programas para OS/2.2.X, UNIX de Sun y USL, son comparables para DOS y Windows 3.X.

UNIX soporta a la mayoría de las aplicaciones familiares de una computadora personal, tales como el Lotus 1-2-3, WordPerfect, y Dbase IV. En realidad se cumple con la disponibilidad de procesadores RISC a un costo mediano, y una alta funcionalidad. Esto ha hecho a UNIX un fuerte contendiente como un cliente para las aplicaciones cliente/servidor. Durante 1994 y 1995 se ha observado que los programas de escritorio (desktop) de multitarea suministrados por Windows NT, Windows 95, UNIX, y OS/2 se convierten en los sistemas operativos seleccionados para el cliente en ambientes cliente/servidor. La selección entre las versiones de Windows, UNIX y OS/2 serán hechas en base al costo de desarrollo más que en la funcionalidad. Previo a la compra, las limitaciones de las PC



animarán a muchas organizaciones a preferir Windows 95 y OS/2 que a Windows NT o UNIX, los cuales podrían requerir adquisiciones nuevas de equipo a parte de tener un mayor costo.

El OSF/1 (un UNIX de grado comercial) está ahora disponible para la plataforma Intel y es la causa de que las organizaciones reexaminen el uso de UNIX en la PC. Los costos de licencia actuales para OS/2 puede darle ventaja al OS/2 si los costos del OSF/1 no son mucho menores a las licencias actuales de UNIX.

El Ambiente Común para Software Abierto (COSE, por sus siglas en inglés) del grupo de los vendedores del núcleo de UNIX han estado de acuerdo en conjunto con los Programas de Interfaz de Aplicación para la mayoría de los servicios de UNIX. Esto permite a los desarrolladores de aplicación construir una aplicación para todas las plataformas. Así servirá para expandir el número de aplicaciones que correrán a través de varias plataformas UNIX. En turno, esto incrementará el uso de UNIX en el escritorio y subsecuentemente reducirá el costo por centro.

El Windows 3.X es por mucho el dominante Interfaz Gráfica de Usuario (GUI) y aun con sus simples limitaciones de tarea, este es el sistema operativo cliente dirigente candidato para las aplicaciones cliente/servidor. Esto proporcionará una plataforma cliente que puede usar mejor las capacidades de la nueva generación de procesadores Intel mientras continua para proporcionar la Interfaz Gráfica de Usuario e Interfaz de Programa de Aplicación de Windows 3.X. Este sistema es probable que gane una parte significativa de la base del cliente usuario en 1996. La complejidad y los recursos de Windows NT sugieren no eliminar muchos programas de escritorio de Windows para la disponibilidad de Windows 95.

### II.1.2 Servicios del Cliente

La plataforma ideal del cliente/servidor opera en sistemas de ambientes abiertos usando una disciplina solicitante/servidor que es basada en estándares bien definidos. Esto permite un equipo múltiple y plataformas de programas para interactuar. Cuando se ha elegido la disciplina cliente/servidor, los servidores pueden crecer y cambiar su sistema operativo y las plataformas del equipo sin cambiar las aplicaciones del cliente. Los clientes pueden estar en máquinas 486/SX, 486/DX, 486/DX2, 486/DX4, PENTIUM o en las poderosas estación de trabajo basadas en RISC, y correr la misma aplicación emitiendo las mismas solicitudes para el servicio mientras se haya elegido la arquitectura cliente/servidor.

Las aplicaciones tradicionales del computador central (host) que usa el cliente para sus servicios de presentación operan únicamente para enviar y recibir caracteres de un servidor. Todas las aplicaciones lógicas residen en el servidor. De esta forma muchas organizaciones usan la tecnología de estación de trabajo hoy en día.

Las aplicaciones de la primera generación del cliente/servidor usan programas tales que permiten la entrada y salida de flujos de datos para ser enviados al cliente sin cambios para las aplicaciones del computador central (host). Utilizan una Interfaz de Programa de Aplicación que define el formato del flujo de datos.

Un ejemplo de esta forma del cliente es una aplicación desarrollada para el comando de emergencia y los servicios de control requeridos por las aplicaciones del escuadrón E911. Esta aplicación de la computadora mantiene las llamadas para el número telefónico de emergencias 911 y para los departamentos de bomberos, policías, ambulancia o para autos en un accidente. Esta aplicación tradicionalmente ha sido implementada en una minicomputadora con acceso proporcionado de una terminal de captura modo carácter. La información es desplegada en forma de lista, y el operador puede mover el cursor a un renglón sobre la lista, para seleccionar o volver a teclear los datos de entrada. Anteriormente las aplicaciones de este programa mantenían la dirección de la persona que hablaba para ser desplegada en pantalla como un campo de texto.

En la implementación del cliente/servidor en este sistema, el usuario de la estación de trabajo trata únicamente con una Interfaz Gráfica de Usuario. La estación de trabajo busca la dirección dentro de un mapa y esta despliega la localización del incendio. En suma, las localizaciones de todas las estaciones de bomberos y los vehículos son buscados en el mapa. El operador del departamento puede echar un vistazo a los estados enteros de soporte para el incendio. Las aplicaciones previas de esta aplicación despliegan listas de vehículos de bomberos opcionales. De la lista, el operador vuelve a teclear un vehículo seleccionado. El límite frontal de la Interfaz Gráfica de Usuario, permite que los vehículos sean mostrados en una ventana y sean seleccionados por medio del mouse. Esto no solo reduce el costo de ejecución, sino que puede reducir significativamente los errores, incrementar la productividad y reducir la tensión experimentada por el operador del departamento.

Las Interfaces Gráficas de Usuario permiten a los usuarios ser más productivos con menor entrenamiento, porque la interfaz es más intuitiva. Varios estudios comparan la productividad y el aprendizaje para los usuarios de las aplicaciones de Interfaz Gráfica de Usuario contra las aplicaciones de modo carácter tradicionales que han mostrado progresos del 200%.

La edición local, la entrada de datos automática, las capacidades de ayuda y otros procesos lógicos pueden ser agregados en el frente de las aplicaciones existentes del servidor. Si muchos errores son detectados en el cliente, o las funciones tales como la ayuda en línea son completamente utilizadas, la carga de trabajo del procesador del servidor disminuye. Existe una oportunidad para proporcionar una ayuda interactiva extensa y un entrenamiento integrado dentro de una aplicación cliente/servidor usando únicamente los servicios de la estación de trabajo del cliente y el Sistema Operativo de Red (NOS).

La conclusión de los archivos impresos frecuentemente implica entradas de datos redundantes dentro de los sistemas múltiples de cómputo o en las aplicaciones. La colección de estos datos en la fuente, dentro de la función de entrada de datos comunes y la

distribución de estos a otras funciones de entradas de datos, puede reducir los costos y errores. Idealmente, la información es registrada por el individuo o por el responsable del proceso para la creación de datos. Esto permite al individuo con el conocimiento de hacer correcciones, hacerlas de inmediato. El servidor del grupo de trabajo LAN captura los datos y los almacena. Cuando un proceso comercial se define para capturar datos de una copia de una forma, los datos almacenados son automáticamente combinados dentro de la forma. Esto es actualizado por el usuario, con datos adicionales que están ahora disponibles. De esta forma, los datos se vuelven a teclear solamente una vez y cada proceso comercial usa los mismos datos. La información está disponible inmediatamente después de capturarla y puede ser distribuida electrónicamente a todos los usuarios autorizados.

Otra técnica usada comúnmente para influir en el poder y la facilidad de uso de la estación de trabajo es proporcionada por las herramientas, tales como los árboles y bosques intrínsecos (Trinzic's Forest). Estas herramientas proporcionan facilidades para la manipulación de datos almacenados en el host de las bases de datos extensa o para la carga en los servidores locales. Esta técnica de "data mining" a través del uso de herramientas de desarrollo poderosas para proporcionar un desarrollo rápido de nuevas funciones de soporte, pronostican el futuro para el desarrollo de sistemas.

### II.1.3 Petición para el servicio

Para los servicios de petición al servidor se necesita saber si este servidor es en realidad el mismo procesador de la computadora o un procesador de la red, para esto el formato de la aplicación de la petición es la misma. El programa del Sistema Operativo de Red (NOS) traduce o suma las especificaciones requeridas para el solicitante hacia la petición de la aplicación.

Los InterProcesos de Comunicación (IPC) son los términos genéricos usados para describir la comunicación entre procesos podrían estar en la misma computadora, a través de la LAN o a través de la WAN.

La mayoría de servicios básicos suministrados por el NOS son redireccionados. Este servicio intercepta al sistema operativo del cliente llamándolos y redireccionándolos al sistema operativo del servidor. De esta forma, las peticiones para los directorios del disco, los archivos del disco, las impresoras, los dispositivos seriales, los programas de aplicación son atrapados por la redirección del programa y redireccionado a una locación correcta del servidor (sobre la LAN). Esto es todavía posible para varios de estos servicios para ser suministrados por la estación de trabajo del cliente. Las unidades del disco local pueden ser designados como A: y C: y las unidades remotas como D:, E: y F:

### II.1.4 Llamadas a Procedimientos Remotos (RPC's)

A través de los años, los buenos programadores han desarrollado un código modular usando técnicas estructuradas y subrutinas lógicas. Ahora, los desarrolladores quieren que

las subrutinas sean almacenadas como objetos nombrados "en cualquier lugar" y hacerlos disponibles para cada uno con el derecho a usarlos. Las llamadas de procedimiento remoto proporcionan esta capacidad. Las RPC's estandarizan la forma en que los programadores deben escribir las llamadas, a pesar de que los procedimientos remotos pueden reconocer y responder correctamente.

Si una aplicación emite una petición funcional y esta petición es empotrada en una RPC, la función emitida puede ser localizada en cualquier lugar en la empresa que el llamador es utilizado a acceder.

La facilidad de una RPC proporciona para la invocación y ejecución de peticiones de los procesadores corriendo en diferentes sistemas operativos y usando las plataformas del equipo del llamador. Muchas RPCs también suministran los servicios de traducción de datos. La llamada causa una traducción de los datos dinámicos entre los procesadores con diferentes formatos físicos de almacenamiento de datos. Estos estándares son desarrollados y han sido adaptados por la industria.

#### **II.1.4.1 Servicios**

##### **Servicios de Fax/Impresión**

El NOS permite al cliente generar peticiones de impresión aún cuando la impresora este ocupada. Estos son redireccionados por el programa redireccionador del NOS y dirigido por el servidor de la impresora. La estación de trabajo del cliente puede ver el estado de la impresora en cualquier tiempo. Muchos servidores de impresión notifican a la estación de trabajo del cliente cuando la petición de impresión es completada.

##### **Servicios de Ventanas**

Una estación de trabajo puede tener varias ventanas abiertas en la pantalla en cualquier tiempo. La capacidad de activar, ver, move, o esconder una ventana en particular es suministrada por los servicios de ventanas del sistema operativo del cliente. Estos servicios son esenciales en la implementación cliente/servidor, porque ellos interactúan con servicios de mensaje proporcionados para notificarle al usuario de los eventos que ocurren sobre un servidor. Cada aplicación es escrita con la suposición que esta tiene una pantalla virtual. Esta pantalla virtual puede ser de un tamaño arbitrario y puede agrandarse como una pantalla física. La aplicación, usando el programa de Interfaz Gráfica de Usuario, pone los datos dentro de la pantalla virtual, y los servicios de ventanas manejan el lugar y la manipulación de la aplicación de la ventana.

## **Servicios de Utilidad**

El sistema operativo proporciona funciones locales tales como, copiar, mover, editar, comparar y ayudar, que se ejecutan en la estación de trabajo del cliente.

### **Servicio de Mensajes**

Los mensajes pueden ser enviados y recibidos hacia una red. Los servicios de mensajes suministran los horarios, el almacenaje y los servicios de vibración para mantener esta función.

### **Servicios de Red**

La estación de trabajo del cliente se comunica con la red a través de un conjunto de servicios y las Interfaces de Programa de Aplicación crean, envían, reciben y formatean los mensajes de la red. Estos servicios proporcionan un soporte para los protocolos de comunicaciones, como el NetBIOS, IPX, TPC/IP, APPC, Ethernet, TokenRing, FDDI, X.25 y SNA.

### **Servicios de Aplicación**

En suma para los servicios de ejecución remota que el NOS proporciona, las aplicaciones de costumbre usarán sus propias interfaces de Aplicación de Programas empotrados en una RPC para invocar servicios especializados de un servidor remoto.

### **Servicios de Bases de Datos**

Las peticiones de bases de datos son hechas usando la sintaxis del SQL. El SQL es un lenguaje estándar dentro de la industria que es proporcionado por muchos vendedores. Porque el lenguaje usa una forma estándar, la misma aplicación puede ser corrida en plataformas múltiples. Hay diferencia de sintaxis y extensiones disponibles del producto para la mayoría de los vendedores.

El uso de procedimientos almacenados es frecuentemente una forma de evitarle al programador el uso de extensiones necesarias para la ejecución.

## **II.1.5 Plataformas**

La plataforma es la combinación de programas y equipo que sirve para que el Manejador de Base de Datos (DBMS, por sus siglas en inglés) cliente/servidor pueda correr. Existen cuatro categorías de plataformas: PC's, Estaciones de Trabajo UNIX (usualmente

RISC), minicomputadoras, y mainframes. A pesar de que la plataforma cliente/servidor más común es en PC's, todas tienen sus ventajas y desventajas.

Mientras los sistemas de equipo varían ampliamente en características y capacidades, necesariamente se necesitan características comunes para el programa del sistema operativo. El sistema operativo es el programa primario que actúa como una Interfaz entre el equipo y la aplicación que corre en este equipo. Las aplicaciones son usualmente escritas para correr bajo un Sistema Operativo en particular. Algunos ejemplos comunes de programas de Sistema Operativo son MS/PC-DOS, OS/2, las muchas variantes de UNIX, DEC's VMS, y el MVS/XA que corren en mainframes IBM.

La primera característica que necesita el Sistema Operativo para DBMS cliente/servidor es multitareas, correr numerosas aplicaciones concurrentemente. Multitareas permite al DBMS el manejo propio de diferentes peticiones de usuarios y respuestas sin que se interfieran por división (tiempo compartido) en el tiempo de procesamiento del CPU entre las diferentes tareas o procesos.

Los sistemas operativos multitareas pueden ser preventivos apropiados o no apropiados. En un sistema apropiado, el Sistema Operativo controla la cantidad de tiempo de CPU para cada tarea. Contrariamente, la aplicación controla el tiempo de CPU en un sistema no apropiado, y sólo se transfiere a otras tareas cuando a finalizado de correr sus tareas. Los sistemas apropiados tienen una natural ventaja para las bases de datos cliente/servidor, estos sistemas se previenen de cualquier tarea que domine completamente el sistema.

El Sistema Operativo también puede ser multiusuario (puede soportar usuarios simultáneamente realizando diferentes tareas), particularmente cuando las terminales tontas son usadas para acceder al DBMS. Un Sistema Operativo multiusuario no tiene una ventaja o desventaja en particular para los DBMS cliente/servidor.

Multihebra es una reciente adición comúnmente disponible en programas de Sistema Operativo apropiado multitareas, mientras el concepto data de los años 70's. Esta capacidad permite a una aplicación multitarea dentro de sí misma, por ejemplo, un usuario multihebra del DBMS puede comenzar una nueva hebra (proceso o tarea) de ejecución para hacer un reporte complejo en el fondo mientras el usuario está consultando el sistema en primer plano. Multihebra tiene implicaciones significativas para el diseño de DBMS's cliente/servidor complejos, así como puede dar un gran control a aplicaciones mientras una nueva tarea es iniciada o detenida. La aplicación puede ser diseñada con inteligencia incorporada para identificar que procesos o tareas tienen una prioridad mayor y proporcionarles más tiempo de CPU que otras tareas. Por ejemplo, el DBMS puede dar una prioridad ligeramente mayor a las modificaciones de datos sobre las consultas de datos, reduciendo la falta de funcionamiento en las que incurren las tareas complejas, sin afectar significativamente la velocidad de consulta.

Ninguna plataforma se ajusta a todas las necesidades o es correcta para cada situación. ¿Cómo se puede elegir la correcta para uno?. Primero se examina el actual sistema DBMS, proyectando su crecimiento, y estimando cuántos usuarios accederán a él al mismo tiempo. Después elegir una posible plataforma en el diagrama de decisión (ver tabla II.1.5.1.1) y platicar con los vendedores.

### II.1.5.1 Principales Plataformas

El presente apartado tiene el Objetivo de describir brevemente las principales plataformas que el mercado nacional ofrece, no se mencionarán todos los productos del mercado, solamente los más representativos por su penetración y que cuentan con el soporte técnico representativo.

Una primera clasificación de los productos comerciales dentro de cada una de las arquitecturas de sistemas operativos, sería:

Arquitectura	Producto	Fabricante
<b>Servidor de Archivos (File Server)</b>	NetWare V. 2.x	Novell
	NetWare V. 3.x	Novell
	Lan Manager	Microsoft
	Lan Server	IBM
	Corestream	Artisoft
<b>Ciente-Servidor</b>	NetWare V. 4.x	Novell
	Windows NT Advanced Server	Microsoft
	Lan Manager	Microsoft
	Redes UNIX	Varios
<b>Punto a Punto (Peer to Peer)</b>	Lantastic V. 6.x	Artisoft
	Windows for Workgroups	Microsoft
	Personal NetWare	Novell
	Windows NT	Microsoft
	MS-DOS	Microsoft

Tabla II.1.5.1.1 Arquitectura, Producto y Fabricante de NOS.

#### II.1.5.1.1 Novell Netware

Novell fué una de las primeras compañías en proveer productos para trabajar en red, estos productos aparecieron cuando predominaban sólo los Servidores de Archivos y de Impresión; "Antes de la aparición del concepto cliente/servidor, Novell ocupaba el primer

lugar en comercialización de productos para red, hoy día Novell ocupa más del 50% de las instalaciones de sistemas operativos para red instaladas mundialmente". Estos antecedentes han provocado que compañías como IBM y DEC provean a sus productos de interfaz para Novell. Novell desde su primera aparición a principios de los 80's estuvo a la vanguardia de los sistemas operativos, entre otras por las siguientes características:

- Fue el primero en soportar múltiples topologías, y conectividad con sistemas IBM, Digital, Apple y UNIX.
- Fue el primero en soportar sistemas operativos como OS/2 y DOS en todas sus versiones.
- Introdujo el Sistema de Tolerancia a Fallas (SFT) y capacidades de disco espejo.
- Novell NetWare no necesita el soporte de otro sistema operativo para el manejo de múltiples tareas (LAN manager y LAN Server, por ejemplo, requieren de OS/2 y UNIX respectivamente).
- El concepto de NLMs introducido por Novell, el cual permite que diversas aplicaciones -sobre todo servidores de bases de datos-, las cuales tienen diversas clases de Interfases de Programa de Aplicación no soportadas por Novell, puedan correr sobre éste como NLM's, los cuales por funcionar como extensiones del sistema operativo se ejecutan con mayor Interfaz de Programa de Aplicación.

#### II.1.5.1.2 Windows NT

Windows NT es un sistema operativo de 32 bits con multitarea con derecho preferente y protección de memoria, que cuenta además con soporte para multiprocesamiento simétrico y trabajo en red, todo ello con una interfaz totalmente gráfica para el usuario.

Las características de trabajo en red local con las que cuenta Windows NT permiten compartir archivos de un sistema con cualquier usuario de un entorno de red, así como conectarse a directorios compartidos de otros sistemas. Las computadoras donde se ejecuta Windows para grupos de trabajo pueden aprovechar las ventajas del trabajo en red. Además, Windows NT viene equipado con el programa y los controladores necesarios para establecer oportunas conexiones con otros tipos de sistemas operativos, como es el caso de UNIX o las grandes computadoras IBM.

El producto **Windows NT Advanced Server** es una versión mejorada de Windows NT que cuenta con características sofisticadas de servidor de archivos para entornos de red de grandes dimensiones. Incluye algunas características adicionales relacionadas con la protección de datos, como es el caso, por ejemplo, de la duplicación automática de datos a discos secundarios.

#### II.1.5.1.3 UNIX

UNIX es uno de los sistemas operativos multiusuario y multitarea de tiempo compartido más famosos del mundo y muy versátil en plataformas para clientes o servidores.



Sin embargo, una de las características que pueden darle más fortaleza en el futuro es el X-Windows System, el cual provee de una efectiva Interfaz Gráfica de Usuario para ambiente Front-End de redes y que es uno de los puntos débiles de UNIX una herramienta de dominio público y muy común. Existe también la posibilidad de construir Arquitecturas cliente/servidor con poderosos servidores de SUN o IBM corriendo bajo UNIX y utilizar herramientas 4GL aunadas a las intrínsecas del shell de UNIX y el lenguaje C.

El Sistema X Windows es realmente un protocolo de comunicación y un conjunto de herramientas de interfaz que utiliza TCP/IP (Transmission Control Protocol/Internet Protocol) Protocolo de control de transmisiones/protocolo Internet.

#### II.1.5.1.4 OS/2

**OS/2** es un sistema operativo **multitarea** que trabaja a 32 bits, desarrollado por IBM y Microsoft, que soporta aplicaciones de cliente/servidor. OS/2 surgió sobre la idea de realizar un proceso downsizing de un sistema operativo de mainframe para el ambiente de microcomputadoras. IBM describe a OS/2 como "Un todo en un sistema operativo" que introduce las siguientes ventajas:

- Un sistema operativo multitarea que trabaja a 32 bits y con procesadores arriba de 386.
- Interfaz Gráfica de Usuario orientada a eventos.
- Iconos de los recursos de la red.
- Ambiente multitarea que protege aplicaciones.

Las **Interfaces Gráficas de Usuario** orientadas a objetos de OS/2 están contenidas dentro del Manejador de Presentaciones (PM) el cual provee de interfaces gráficas de programación que derivan de los sistemas de mainframes. Este manejador trabaja por eventos (click's al mouse, entradas de teclado etc.) y se basa en transmisión de mensajes que se asocian con ventanas; existen herramientas 3GL y 4GL que soportan este ambiente y lo cual hacen que sea accesible al mercado. Para emplear a OS/2 como servidor IBM introdujo su programa LAN Manager y NetWare. IBM ha logrado incorporar a través de LAN Server una amplia variedad de funciones de red tales como: acceso remoto a los recursos y aplicaciones de red, manejo y administración de red, controles sobre accesos y seguridad, protocolo de comunicaciones y soporte a programación de Interfaz de Aplicación de Programa, además de brindar la posibilidad de que programas como NetWare corran sobre OS/2. la principal ventaja que encontramos en las características antes mencionadas, es sin duda la posibilidad de llevar a cabo un adecuado downsizing de mainframes sobre plataformas de PC's en una Arquitectura cliente/servidor mediante un protocolo de stacks (que proveen conectividad con Ethernet LAN, NetBios, TCP/IP, X.25 y SNA/SKLC, en este caso OS/2 funge como puerta de acceso (gateway) con un mainframe y comunicación de Interfaz de Aplicación de Programa.

Algunos sistemas operativos -Arisoft LANtastic por ejemplo- están diseñados para trabajar en pequeños grupos, los cuales en ocasiones consisten solo de algunas PC's

conectadas en red; sin embargo los sistemas operativos más comúnmente usados en una Arquitectura cliente/servidor son Novell Net Ware, Microsoft LAN Manager, IBM LAN Server y Banyan Vines. Estos sistemas operativos tienen la capacidad de atender a un mayor número de usuarios brindando sofisticados requerimientos operacionales.

### II.1.5.1.5 LAN Manager

LAN Manager y LAN Server son productos desarrollados por Microsoft e IBM respectivamente, pero que conservan características similares principalmente debido a que se desarrollaron en conjunto y los dos trabajan con OS/2.

LAN Manager es usado en Arquitecturas cliente/servidor que tienen una base de OS/2 en los servidores y conserva las siguientes características:

- Al trabajar OS/2 en conjunto con LAN Manager permite que el sistema operativo sea multiusuario y no sólo multitareas.
- LAN Manager se adapta sin problemas a cualquier tamaño de red.
- Soporta conexión con Novell NetWare.
- Soporta diversas arquitecturas como Token Ring, Ethernet, FDDI y Arcnet.
- Soporta diversos protocolos como TCP/IP, Net Bios y Named Pipes.

### II.1.6 Protocolos de la comunicación

Los protocolos usados en la Arquitectura cliente/servidor emplean una o varias técnicas para el envío de mensajes. Esta sección hace referencia a protocolos usados en cliente/servidor particularmente:

- NetBios
- Named Pipes
- DCE-RPC
- APPC de IBM
- TCP/IP

#### II.1.6.1 NetBios (Network Basic Input/Output System)

Es un protocolo comúnmente usado como una Interfaz de Programa de Aplicación para ambientes de redes locales. NetBios actúa como una Interfaz entre las capas de transporte y de sesión del modelo OSI, las cuales son definidas por el protocolo Net BEUI. NetBios funciona como un protocolo de transmisión full-duplex que soporta la transmisión de datos por circuitos virtuales y servicios connection oriented y connection less oriented. NetBios corre sobre MS-DOS y OS/2, es soportado por TCP/IP y es posible trabajarlo con productos como SLAN Manager y Novell Netware. Así mismo NetBios puede soportar:

Token Ring, Ethernet y Arnet por lo que puede ser considerado como un posible estándar para aplicaciones de redes.

### **II.1.6.2 IBM APPC (Application Program-to-Program Communication)**

Los APPC forman parte de un protocolo recomendado por IBM para IPC en la arquitectura SNA de IBM, que permiten comunicaciones punto a punto entre cualquier programa en un ambiente de esta arquitectura. Las entidades en este ambiente usan APPC para realizar funciones de comunicación que son llamadas por los programas de transacción (TP). Un programa puede contener uno o más TP's, cada nodo de la red de una arquitectura SNA contiene una Unidad Física (PU) la cual maneja los buffers de datos y las ligas de la red. Esta unidad física determina las funciones provistas por los nodos, estas funciones pueden ser de ruteo, conexión o direccionamiento de dispositivos.

Otro elemento de este protocolo es una Unidad Lógica (LU) que provee el acceso de los Tp's a la red, muchas unidades lógicas puede actuar simultáneamente o bien muchos Tp's pueden acceder los servicios de una unidad lógica. Una de las facilidades y ventajas que presenta este protocolo es la posibilidad de que dos nodos de la misma, cualesquiera que sean pueden comunicarse independientemente de su localización en la red, del sistema operativo o de la plataforma de equipo:

De esta forma una estación de trabajo que opere como MS-DOS puede establecer comunicación con un mainframe IBM provistos ambos de programas de soporte.

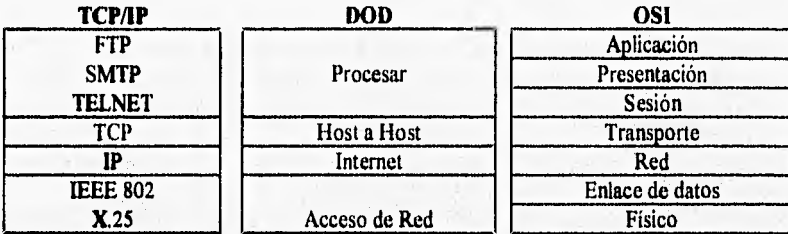
### **II.1.6.3 TCP/IP (Transmission Control Protocol/Internet Protocol)**

TCP/IP (Transmission Control Protocol/Internet Protocol).Protocolo de control de transmisiones/protocolo Internet.

Conjunto de protocolos de comunicaciones desarrollado por la Defense Advanced Research Projects Agency (DARPA - Agencia de proyectos de investigación avanzada de defensa) para intercomunicar sistemas diferentes. Se ejecuta en un gran número de computadoras VAX y basadas en UNIX, y es utilizado por muchos fabricantes de equipo, desde los de computadoras personales hasta los de macrocomputadoras. Es empleado por numerosas corporaciones y por casi todas las universidades y organizaciones federales de los Estados Unidos.

El File Transfer Protocol (FTP - Protocolo de transferencia de archivos) y el Simple Mail Transfer Protocol (SMTP -Protocolo simple de transferencia de correspondencia) brindan capacidades de transferencia de archivos y de correo electrónico. El protocolo TELNET proporciona una capacidad de emulación de terminal que permite al usuario interactuar con cualquier otro tipo de computadora de la red. El protocolo TCP controla la transferencia de los datos, y el IP brinda el mecanismo para encaminarla.

En el siguiente diagrama se muestran TCP/IP, junto con los modelos DOD y OSI.



El Protocolo de Control de Transmisiones/Protocolo Internet (TCP/IP, por sus siglas en inglés) fue desarrollado originalmente en los Estados Unidos de Norteamérica para ser usado en la red Arpanet (ahora conocida como Internet) la cual unía a varias universidades y centros de investigación en ese país, hoy es la red más grande del mundo. Este antecedente, aunado a que no se burocratizó por ninguna organización, hizo posible que TCP/IP hasta la fecha sea uno de los protocolos de comunicación más aceptados internacionalmente.

**TCP/IP es un protocolo que permite:**

- **La comunicación entre aplicaciones** ubicadas en equipos con características diferentes como marca, sistema operativo, etc.
- **El envío de mensajes** sin necesidad de conexión física directa entre computadoras.
- **El transporte de datos** de manera confiable debido a que TCP/IP se encarga de la recuperación automática de errores de transmisión, pérdida de paquetes de datos o fallas en puntos intermedios a lo largo de la trayectoria que sigue la información de transmisor a receptor.
- **El ruteo de la información** hasta el destino final, sin necesitar una conexión directa entre origen y destino.
- **Al apoyarse TCP/IP** de los protocolos de comunicaciones que se utilizan a nivel capa 2 (Enlace de Datos), permite que se pueda utilizar X.25 o Ethernet.
- **Conectividad universal**, debido a que cada computadora dentro de una red con TCP/IP es reconocida universalmente por medio de una dirección que la identifica.

TCP/IP esta formado por los siguientes protocolos:

**a)IP (Internet Protocol).** Es un protocolo no orientado a conexión que brinda el servicio de envío de paquetes a TCP y otros protocolos. IP contiene una serie de direcciones de 4 bytes que cumplen con las siguientes funciones:

- Identificar de manera única a cada nodo de una red o de un grupo de redes. Identificar a miembros de la misma red (por tener el mismo prefijo).
- Poder dirigir o direccionar información entre un nodo y otro, aún cuando estén en distintas redes.
- Poder dirigir o direccionar información a todos los miembros de una red o grupo de redes.

De esta forma si un paquete tiene que atravesar diversas redes para llegar a su destino, IP consulta su tabla de direcciones tanto de redes como de grupos para determinar donde debe mandarse esta información. Sin embargo IP no cuenta con mecanismos que garanticen la llegada de paquetes o el control del flujo. Estos mecanismos están contemplados en TCP.

**b)TCP (Transmission Control Protocol).** Es un protocolo orientado a conexión que permite y garantiza el envío de paquetes (datos) a los procesos del usuario y regula el flujo de la información. Esto significa que cada mensaje que un nodo desea enviar, se partirá en paquetes a los cuales se les asignará un número. Para completar la tarea anterior, cada nodo que recibe uno o varios paquetes, le informa al que está enviando que efectivamente, si los ha recibido. Toda la información del número de secuencias del paquete, del reconocimiento, del control de flujo y otros datos, esta contenido en el encabezado de TCP.

Estas tareas son posibles gracias a que antes de que se envíe la información entre dos nodos, TCP establece una conexión, de igual forma cuando se ha terminado de enviar paquetes, se hace el proceso contrario para terminar la conexión satisfactoriamente.

#### II.1.6.4 Sockets

“Un socket es un objeto abstracto desde el cual los datos son transferidos. Los sockets existen dentro de un dominio de comunicación y cambian datos con otros sockets residiendo en el mismo dominio”. Unix usa el dominio de los sockets para comunicarse entre procesos en un host, considerando el dominio de sockets Internet para soportar la comunicación entre procesos hosts separados.

Los sockets así mismo, son un importante componente de una red con TCP/IP y parte de la base para construir el sistema de E/S en una red que opere en un ambiente UNIX.

Una red con TCP/IP puede contener un gran número de sockets, pero cada socket está identificado con una aplicación específica o nodo específico de la red.

Actualmente las aplicaciones cliente/servidor requieren nuevas demandas de los desarrolladores, incluyendo la necesidad del manejo de interproceso de comunicación (IPC). "Este interproceso de comunicación es la transferencia de datos entre programas que están o no localizados en la misma computadora". Más facilidades de IPC están diseñadas para transferir datos entre objetos a otros procesos.

Ahora se tratarán otros componentes de la Arquitectura cliente/servidor:

### **II.1.7 Interfaces Gráficas de Usuario e Interfaces de Programas de Aplicación**

**Interfaz Gráfica de Usuario.** Es una de las mayores ventajas que ofrece una Arquitectura cliente/servidor ya que brinda la posibilidad de manejar interfaces gráficas entre los usuarios y las aplicaciones, facilitando la comprensión, el manejo y el aprendizaje de las mismas. Otra de las características que hacen de las Interfaces Gráfica de Usuario herramientas de trabajo poderosas, es la posibilidad de correr diversas aplicaciones al mismo tiempo y navegar a través de éstas, abriendo y cerrando ventanas; sin mencionar la facilidad que resulta trasladar archivos con sólo arrastrarlos al directorio deseado.

Las aplicaciones basadas en Interfaz Gráfica de Usuario, pueden proveer de un gran número de operaciones de una manera más amigable para los usuarios. En una aplicación basada en Interfaz Gráfica de Usuario se pueden hacer múltiples diseños de pantallas donde operan menús que se accesan con un dispositivo de puntero (llamado comúnmente mouse), botones de ayuda o de acceso a aplicaciones; con la posibilidad de mantener varias aplicaciones al mismo tiempo y transportar datos y archivos de manera más fácil y cómoda.

La tecnología más conocida en este ambiente es **X-Windows**, con ésta se pueden desplegar simultáneamente múltiples aplicaciones usando una o más ventanas. X-Windows está basado en la Arquitectura cliente/servidor proveyendo de un protocolo de comunicación entre una aplicación y su lógica de presentación. Esto es particularmente exitoso en un ambiente heterogéneo, donde PC's, estaciones de trabajo y otras plataformas pueden correr las mismas aplicaciones.

Las aplicaciones hechas para un Interfaz Gráfica de Usuario no son portables para otros ambientes de Interfaz Gráfica de Usuario. Y para satisfacer esta necesidad sería necesario desarrollar un estándar que pudiera operar independientemente del sistema operativo o de la plataforma de equipo que se tenga y en caso de redes, independientemente del protocolo que se emplee. Sin embargo, hoy en día el estándar de Interfaz Gráfica de Usuario es X-Windows, el cual ha servido como modelo para la ANSI y el Instituto de los estándares IEEE:

Otro concepto importante dentro de las Interfaces Gráficas de Usuario son las **Interfaces de Programas de Aplicación**. "Una Interfaz de Programa de Aplicación es un método por el cual el desarrollador referencia los servicios que están disponibles en el ambiente y es utilizado para crear la parte frontal de las aplicaciones con respecto a los servicios de la red". Una Interfaz de Programa de Aplicación integra un conjunto de

funciones y llamadas a funciones que los programadores usan para acceder a los servicios antes mencionados, Interfaz Gráfica de Usuario pueden ser la creación y ajuste de una ventana, el llamado de una base de datos. Para la construcción de una Interfaz de Programa de Aplicación e recomiendan lenguajes nativos como SDK de Microsoft, aunque existen otras herramientas con las cuales realizar esta tarea como Presentation Manager de IBM o inclusive en lenguaje C y ensamblador.

Una Interfaz de Programa de Aplicación contiene funciones (que realizan tareas como manejo de ventanas, translación de cadenas, creación de sonidos, mensajes, etc), estructuras de datos, tipos de datos y archivos. Todos estos elementos permiten que se puedan crear programas que corran en un ambiente Windows. La Interfaz de Programa de Aplicación está compuesta por tres elementos básicos:

a) La descripción del servicio. Especifica el propósito del servicio y cómo será usado a través de especificaciones de un lenguaje como C, o C++.

b) Tipos de Datos. Son los tipos de datos comunes o definidos por el usuario que se emplearán en las funciones utilizadas.

c) Llamadas a Funciones. Aquí se declaran las funciones que se envían y reciben información.

### II.1.8 Las Aplicaciones

Para comprender mejor los límites del mainframe así como del servidor de archivos, dividiremos una aplicación en sus partes componentes:

- Lógica de presentación
- Lógica del negocio
- Lógica de datos
- Servicios de Comunicaciones

**Tabla II.1.8.1 Partes componentes de una aplicación basada en cliente/servidor**

#### **Lógica de presentación**

Está provista de un dispositivo que acepta la entrada desde el usuario y despliega cualquier presentación lógica que se llame. Este puede ser un simple dispositivo de carácter

con sólo el teclado de entrada o uno más complejo como las imágenes que pueden aceptarla desde un **mouse**. Controla la interacción entre el usuario y la computadora. Esto es la especificación de qué pasa si el usuario selecciona una opción del menú, presiona un botón, o elige un elemento (**ítem**) de una lista?

### **Lógica de negocio**

Es un conjunto de decisiones, cálculos y operaciones que la aplicación debe sacar. Esta puede incluir el cálculo de las compensaciones de empleados, la decisión de aceptar una orden, la evaluación de la aplicación de un préstamo, o los procedimientos tras una transacción tales como una transferencia de fondos.

### **Lógica de datos**

Es la expresión de las operaciones a ser transformadas en la base de datos que es necesaria para sacar la lógica del negocio. Para las bases de datos relacionales estas expresiones están en sentencias SQL tales como SELECT, UPDATE, INSERT. Son las acciones que toma el DBMS para extraer la lógica de datos, incluyendo manipulación y/o definición de datos, transacción commit/rollback, etc. Para la manipulación de datos el DBMS típicamente compila en SQL.

### **Servicios de comunicaciones**

Es la parte que se encarga del envío y recepción de los paquetes de información que viajan a través del protocolo de comunicaciones que se establece al usar un medio físico (cable, concentrador, ruteador, etc.).

## **II.1.9 Seguridad**

La seguridad también es un elemento importante de la Arquitectura cliente/servidor, por lo que a continuación hablaremos de los elementos que la conforman.

La seguridad en un mundo distribuido es tarea difícil. La responsabilidad descentralizada puede ser la perdición tanto como un punto único de control puede ser un punto único de falla.

### **II.1.9.1 Proceso de Autenticación**

Existen tres métodos diferentes por medio de los cuales se puede verificar la autenticidad del usuario: usando algo que conozcan, algo que tengan o algo que sean. Estos métodos se describen detalladamente a continuación.



El método de "algo que sepan" está tipificado por el uso del identificador (el nombre de su cuenta) y la contraseña del usuario, o por los sistemas que algunas veces preguntan cierta información personal.

La idea es que este conocimiento no está por escrito, por lo que no es probable que esté al alcance de un intruso. Desafortunadamente, los ID (identificadores) pueden obtenerse fácilmente, y algunas veces las personas conservan sus contraseñas por escrito e incluso las tienen pegadas en sus monitores.

Con el método de "algo que tienen", que añade un segundo nivel de confidencialidad al proceso de Autenticación, puede solicitarse al usuario que tenga en su poder algún objeto físico con el fin de obtener acceso (el equivalente electrónico a una llave). Aunque este objeto puede ser tan simple como una tarjeta de plástico con banda magnética, la solución más común actualmente toma la forma de un generador de contraseñas aleatorias o un dispositivo de reto/respuesta.

**Dispositivo de Tipo Ficha:** Un tipo de ficha, como son llamados estos dispositivos, proporciona una palabra alfanumérica pseudo-aleatoria que cambia cada minuto más o menos, y que está sincronizada con una base de datos. Esto da como resultado una contraseña que sólo sirve en un momento específico de tiempo y sólo para entrar en sesión una vez.

El nivel final (y el más confiable) de Autenticación, el método de "algo que sean", involucra algún aspecto único e infalsificable de la anatomía de una persona; esta nueva tecnología aparentemente puede leer patrones faciales infrarrojos al paso de las personas usando sólo una simple cámara de video para capturar las imágenes.

**Los sistemas biométricos** ofrecen el más alto grado de confiabilidad de que el usuario sea realmente quien dice ser, pero también son generalmente los más costosos de implantar.

### II.1.9.2 Kerberos

Kerberos es un sistema basado en encriptamiento diseñado para autenticar usuarios y conexiones de una red. Kerberos tiene a su cargo evitar los accesos no autorizados. Lo hace tan bien, que ahora es casi estándar para efectuar comunicaciones seguras y auténticas a través de una red.

Sin embargo, no obstante su popularidad, Kerberos no es una solución de seguridad completa, aunque proporciona autenticación de usuarios, no maneja la autorización de aplicaciones ni de transacciones dentro de las aplicaciones. Cualquier determinación de autorizaciones de acceso y de derechos debe ser manejada por otros sistemas de la red.

### II.1.9.3 Una sola conexión

Para los ambientes actuales un usuario probablemente necesite tener acceso a varios mainframes, a una red corporativa, a una o más LAN, a sistemas especiales de desarrollo,

etc. Por esto un solo usuario puede acabar teniendo siete, ocho o más contraseñas. Pero allí es donde fracasa completamente el sistema de contraseñas. Nadie puede recordar 15 contraseñas, todas diferentes, difíciles de adivinar y que no se encuentren en el diccionario. ¿No sería lo ideal que uno se pudiera conectar a un solo sistema, y que por el resto de esa sesión, todos los demás sistemas o redes a los que necesitaríamos conectarnos, consultaran una base de datos de seguridad para determinar nuestros derechos, sin necesidad de más procesos de entrada en sesión, interrupciones o contraseñas?. Esto es la unisección.

El principal problema una vez más es consecuencia del ambiente multiplataforma, multi-sistema-operativo y multiprotocolo, en el cual es sumamente difícil poner en práctica esta unisección.

#### II.1.9.4 Confidencialidad

Aún cuando los controles de acceso sean estrictos y bien mantenidos, se necesita otro mecanismo de control para asegurar que la información de la organización conserve su confidencialidad al ser trasladada a lo largo de los extremos de un sistema distribuido, o hacia otras redes.

Mientras que un sistema envíe datos legibles entre usuarios y servidores, será vulnerable al espionaje sobre la red. La solución a este problema es bien conocida: encriptamiento. Esta sólo transforma la señal para que cualquiera que la intercepte, sin importar cómo lo hagan, simplemente no la pueda leer. El encriptamiento es relativamente barato y puede ser bastante fácil de usar.

Los problemas con el encriptamiento yacen en el intercambio y en la administración de las llaves del mismo. Por ello se inventó el método de llave pública, con este se da vuelta al problema de la administración de las llaves, usando una llave para el encriptamiento y otra para el desencriptamiento.

Cualquiera que sea el tipo de seguridad que se tenga instalado en los sistemas distribuidos, se tiene que administrar: agregar nuevas personas, eliminar a las que se han ido, modificar derechos y permisos conforme van cambiando los requerimientos del trabajo.

Lo que ha pasado al irse volviendo distribuidos los sistemas y la información es que diferentes departamentos son responsables de diferentes plataformas. La administración en el mainframe es manejada por un grupo diferente del que maneja las LAN, y de hecho, quizá sean diferentes personas las que administran cada LAN. Y en la mayoría de los casos, la seguridad no es manejada por el departamento de seguridad sino por los departamentos usuarios, donde el administrador de la LAN es también el administrador de seguridad.

La responsabilidad por la seguridad de las comunicaciones también está distribuida similarmente. Anteriormente todas las comunicaciones llegaban a una ubicación central, donde un personal bien capacitado sobre la red las monitoreaba. Ahora llegan a cualquier punto de la corporación, sin ninguna administración central.

La seguridad es en gran parte una cuestión de administración, y no es un problema técnico. La primera cosa que se tiene que lograr es convencer de la importancia de contar con un programa de seguridad, con una política general corporativa y con un conjunto de estándares. Obviamente, diferentes sistemas necesitan diferentes niveles de seguridad.

#### **II.1.9.5 Sistemas Operativos y seguridad.**

Otro factor que influye en la seguridad es la variedad de sistemas operativos a escoger, particularmente de sistemas operativos de red. Algunos tienen más seguridad que otros, pero el problema principal, desde la perspectiva de una red, es la necesidad de dar cabida a múltiples sistemas operativos. Esto genera problemas de compatibilidad y de consistencia.

Es necesario que los sistemas operativos tengan un considerable impacto sobre la seguridad de un sistema distribuido como se muestra en la siguiente tabla:

CONCIEN- TIZACIÓN	ACCESO	AUTENTI- FICACIÓN	CONFIDEN- CIALIDAD	ADMINIS- TRACIÓN	DISPONI- BILIDAD
Establezca una política organizacional con respecto al acceso a las computadoras, a la propiedad de la información y a las responsabilidades del usuario.	Tome precauciones extra si va a permitir el acceso por marcaje (use, por ejemplo, modems de retrolamada y permita el acceso externo sólo a través de un sistema de paredón.	Las contraseñas deben expirar a intervalos regulares.	Encripte los datos transmitidos sobre la red.	Usted debe poder administrar la seguridad (para agregar, eliminar o revisar las autorizaciones de acceso o las listas de control de acceso) desde una estación de trabajo en cualquier parte de la red.	Prepare a sus servidores contra las interrupciones de energía con fuentes ininterrumpibles de poder.
Conciencie a los usuarios de su responsabilidad personal de proteger la información de la organización		Use fichas o tarjetas inteligentes como generadores aleatorios de contraseñas.	Encripte los archivos almacenados que contengan datos especialmente delicados o críticos.		Asegúrese de que todos los datos estén respaldados y que sean recuperables en línea, por ejemplo, vái RAID o intercambio inmediato.

Tabla II.1.9.5.1 Seguridad en un Sistema Distribuido

## II.2 PROGRAMA DE CONEXIÓN

Algunos autores manejan este concepto para referirse a los programas que accesa a los datos mientras que otros lo usan como el programa que conecta dos módulos, permitiéndoles comunicarse entre sí. De una forma más formal podemos definir al Middleware como un conjunto de servicios que hace posible que los desarrolladores de aplicaciones cuenten con Interfaces de Programas de Aplicación que facilitan el desarrollo de aplicaciones. Por ejemplo, para un módulo del cliente que necesita comunicarse con un módulo del servidor algunos autores usan el término "middleware" para referirse al programa que realiza esta función. La parte del middleware esta contemplada en el modelo OSI, en las capas 5,6 y 7.

La presencia y utilización del middleware provee de las siguiente ventajas en un sistema distribuido:

- Acceso a servicios de aplicación disponibles en la red. Los servicios de aplicación son aquellos que proporcionan servicios comunes de información o capacidades de procesamiento para múltiples aplicaciones, tales servicios pueden ser:
  - a) Servicios de Mensajes. Los cuales proveen la habilidad para enviar y recibir mensajes entre y a través de las aplicaciones de los usuarios finales.
  - b) Servicios de acceso a base de datos. Los cuales permiten consultar y recibir información de bases de datos locales o remotas.
  - c) Servicios de Procesamiento Distribuido. Estos permiten que el procesamiento se distribuya a través de la red por medio de RPCs o estructuras de SQL.
  - d) Servicios de Directorio. Los cuales constituyen un mecanismo para la identificación de usuarios, aplicaciones y sistemas y ayudan a simplificar tareas de direccionamiento.
  - e) Servicios de Seguridad. Los cuales brindan autenticidad, control de acceso y confidencialidad para las aplicaciones y otros servicios del middleware.
- Las aplicaciones que son desarrolladas con interfaces consistentes de middleware son portables y reusables a plataformas que conserven esas interfaces, además de ayudar a reducir el tiempo de desarrollo de las mismas.
- Propiamente diseñadas e implementadas, el middleware provee acceso a los servicios integrales de la red requeridos por las aplicaciones. La integridad de estos servicios constituye la clave para reducir la complejidad de los sistemas distribuidos.

Para redondear el tema sobre la arquitectura Cliente /Servidor a continuación se muestra una tabla comparativa entre el modelo tradicional y dicha arquitectura.

**VENTAJAS DE CLIENTE/SERVIDOR SOBRE SISTEMAS TRADICIONALES.**

<b>SISTEMA TRADICIONAL</b>	<b>CLIENTE/SERVIDOR</b>	<b>VENTAJAS DE CLIENTE/SERVIDOR</b>
Centralizado	Distribuido	Inteligencia y proceso distribuido. Disponibilidad rápida de información crítica. Competitividad
Sistema cerrado	Sistema abierto	Portabilidad de datos, aplicaciones. Libertad del comprador
Mainframe, Minis, terminales de Host	Pc's, LANS, Interfaces gráficas estaciones de trabajo y servidores RISC/Unix, RDBMS	Precio/rendimiento. Mejor valor. Costo
Base de datos jerárquico	Base de datos relacional y destibuida	Flexibilidad y facilidad de acceso a información.
Interfaz de base de datos propietaria	Estándares empresariales para bases de datos relacionales	Acceso universal
Lenguajes de 3a. generación	Gráficas/Orientadas a objetos, lenguajes de 4a. generación y herramientas CASE	Configurabilidad y modularidad de Interfaz de Programa de Aplicación.
Interfaz para el usuario de caracteres	Interfaz gráfica de usuario	Facilidad de uso y capacitación. Productividad
Soporte de sistema operativo único	Estándares de interfaces para programación de aplicaciones Interfaz de Programa de Aplicación.	Portabilidad de sistemas abiertos.

### II.3 SERVIDORES

El servidor es la conjunción de programas y equipo que responde a los requerimientos de los clientes. Como es bien sabido, muchos de los servidores para redes de área local basados en Intel vienen con Microsoft DOS y Microsoft Windows, esto significa que con éste programa sólo son máquinas personales. Habría que instalar algún sistema operativo de red o un sistema operativo para trabajo en grupo, a fin de que sean efectivamente aprovechados como servidores.

Considerando la función que realizan, los servidores se clasifican en los siguientes tipos:

- a) Servidores de archivos
- b) Servidores de datos
- c) Servidores de cómputo
- d) Servidores de aplicaciones
- e) Servidores de bases de datos
- f) Servidores de comunicaciones

Un servidor de archivos es aquel que permite compartir archivos entre un grupo de trabajo. Cuando un cliente solicita un archivo, el servidor envía una copia completa al cliente, incluyendo los índices del archivo. Al suceder esto, se bloquea el archivo hasta que el cliente lo libera, en el mejor de los casos el bloqueo se hace por un rango de registros.

Los servidores de datos y los servidores de cómputo se emplean conjuntamente, el servidor de datos se encarga de proporcionar los datos que el servidor de cómputo le solicita. El servidor de datos no efectúa ningún tipo de procesamiento incluido en la aplicación, sólo hace validación de datos basado en las reglas del negocio. Así, es el servidor de cómputo es el que ejecuta la aplicación. La ventaja de tener estos dos tipos de servidores por separado es que se puede tener uno de ellos más poderoso que el otro, en función de los requerimientos de la aplicación.

El servidor de aplicaciones es aquel que efectúa tanto el manejo de los datos, como el procesamiento de la información, o sea, es un servidor de datos y un servidor de cómputo en la misma máquina.

En la cliente/servidor el caso más común de servidor es el de base de datos. En el corre el manejador de base de datos, generalmente en un RDBMS, que efectúa las funciones de actualización, consulta, verificación de integridad referencial, administración de recursos, etc.

Los servidores de comunicaciones adquieren dos formas, una computadora general con programas específicos para comunicaciones o servidores de comunicaciones como equipo de comunicaciones específico. En ambos casos, los servidores de comunicaciones

sirven principalmente para efectuar funciones de control de acceso, y enrutamiento y puenteo de tráfico.

### II.3.1 Red de cómputo

La red de cómputo es el equipo y programas de comunicaciones que enlaza a los clientes con los servidores, de ahí su importancia en la arquitectura cliente/servidor.

Dependiendo de su cobertura, las redes se clasifican en redes de área local (LAN), redes de área metropolitana (MAN) y redes de área amplia (WAN). Difieren principalmente en la distancia que pueden cubrir, la tecnología de comunicaciones que emplean, el tipo de equipo, los canales de comunicación que pueden usar y la velocidad a la que pueden operar.

Así, por ejemplo, mientras que para una LAN se puede emplear las tecnologías Ethernet (10 Mbps), Token Ring (4 ó 16 Mbps), FDDI (100 Mbps), CDDI (100 Mbps) y ATM (44.736 Mbps), para una WAN es más recomendable ATM y Frame Relay (2.048 Mbps) aunque también se puede emplear Ethernet. Cabe destacar que la tecnología de red no sólo determina el ancho de banda del canal, sino también la forma en que se emplea; tal es el caso de Frame Relay que utiliza el principio de ancho de banda bajo demanda.

Bajo una tecnología de comunicaciones se pueden emplear diversos protocolos. Por ejemplo, en una red Ethernet se pueden emplear los protocolos TCP/IP, NetBios, IPX/SPX, etc. De esta manera, mientras que para una red de área local pequeña o una subred puede resultar adecuado IPX/SPX, para una red de área local con muchos servidores o muchas subredes, o inclusive una red de área amplia, resulta mejor TCP/IP.

En todos los casos lo importante es que la red sea un medio eficiente y confiable para que la comunicación entre clientes y servidores no sea un cuello de botella.

Es importante enfatizar que la arquitectura cliente/servidor ha influido de tal manera en la tecnología de redes de cómputo que los principales fabricantes de equipos de red ofrecen productos especiales para cliente/servidor. Muchos de ellos presentados en forma de concentradores conmutados (switching hubs), o la nueva tendencia, las redes virtuales (virtual networks).

### II.3.2 Distribución entre el cliente y el servidor

Como ya se mencionó, tres son los elementos a distribuir entre el cliente y el servidor: el manejo de datos, la aplicación y la presentación.

En los extremos se encuentran los casos más frecuentes:

- Presentación en el cliente, aplicación y manejo de datos en el servidor
- Presentación y aplicación en el cliente, manejo de datos en el servidor



Por ejemplo, el primer caso se obtiene cuando el cliente corre un paquete de emulación de X terminal para operar bajo X Windows y de esa manera obtener presentación gráfica Motif, mientras que la aplicación y la base de datos residen en el servidor. Aquí la capacidad de cómputo del cliente se destina a proporcionarle la capacidad de interfaz gráfica de usuario Motif.

En el segundo caso, sólo el manejador de base de datos reside en el servidor, mientras que la aplicación y el sistema de presentación residen en el cliente usualmente Microsoft Windows. Aquí, se aprovecha de mejor manera la capacidad de cómputo del cliente, a la vez que descarga de parte de trabajo al servidor, pues la aplicación ya no corre en él.

Los tres casos restantes son puntos intermedios entre los extremos mencionados: presentación distribuida entre cliente y servidor, aplicación distribuida entre cliente y servidor y manejo de datos distribuido entre cliente y servidor.

Ninguno de los cinco casos es mejor que los cuatro de manera absoluta, depende del caso específico, ya que en muchas ocasiones son realmente pasos intermedios para pasar de un tipo a otro. Finalmente, hay un caso que no deja de ser singular a pesar de ser frecuente: que el cliente y el servidor residen en la misma computadora. Por ejemplo, se tiene una aplicación cliente/servidor corriendo en modo carácter en terminales tontas o en microcomputadoras. La ventaja de este modo de operación es que se pueden emplear terminales baratas o microcomputadoras con pocos recursos.

### II.3.3 Interfaz de usuario (Front end) - Sistema de fondo (Back end)

Cliente/servidor contempla a nivel de funciones dos elementos, una llamada Front End, y la otra Back End.

**Front End** : Es la parte de la aplicación que interactúa con el usuario, está ubicada en la máquina cliente, está orientada a la interpretación y presentación de la información procedente del servidor, generalmente está conformada bajo un ambiente gráfico, de ahí que recalquemos la importancia de la capacidad del cliente para el manejo de Interfaz Gráfica de Usuario.

**Back End** : Son los módulos que residen en el servidor, básicamente orientados a recibir solicitudes del cliente con capacidad de procesamiento y de regresar la información solicitada a éste.

Hay que destacar que esta arquitectura se encuentra particularmente provista para el manejo distribuido de datos, debido a que rompe las aplicaciones lógicamente para lograr un procesamiento más rápido y eficiente.

Los Sistemas Manejadores de Bases de Datos Relacionales (RDBMS) juegan un rol muy importante en la arquitectura cliente/servidor ya que aseguran la integridad de la información y la seguridad de la misma, a la vez que proveen el control y la disponibilidad necesarios para aplicaciones de producción en línea. Así mismo, los RDBMS manejan la recuperación de los datos.

Consideremos al Servidor como una computadora que provee a los clientes de elementos como discos de gran capacidad, bases de datos y/o conexiones a la red. Es importante señalar que hablamos de servidores inteligentes. Los servidores pueden ser mainframes, minicomputadoras o potentes estaciones de trabajo.

En el Back End las aplicaciones efectúan actividades que requieren un consumo importante de recursos de procesamiento, mejorando ampliamente el poder de proceso de la red. Explotando la inteligencia del servidor, se puede contar con una administración de datos más controlada, una mejor seguridad y una más fácil administración de la red. La diferencia de un Mainframe es que en esta Arquitectura cliente/servidor la Interfaz del usuario, así como otros procesos específicos de la aplicación, se efectúan del lado del cliente.

El servidor también debe efectuar toda la labor intensiva de comunicación con otros ambientes, redes y topologías.

#### **II.3.4 Consideraciones para la implementación de servidores**

- Debe hacerse un análisis previo de las aplicaciones existentes, de los requerimientos del usuario, del tamaño y los recursos con que cuenta la empresa.
- Determinar cuántos servidores y clientes son necesarios para soportar las aplicaciones que está requiriendo la empresa.
- En qué lugares se necesitan establecer estos clientes y estos servidores.
- Establecer máximos y mínimos de soporte

En la llamada Arquitectura de Servidor de Archivos todo el procesamiento de aplicaciones ocurre en el cliente, y el servidor sólo cubre los requerimientos que estos le soliciten. En esta arquitectura los usuarios pueden requerir el acceso a los archivos del servidor de archivos y el servidor está conectado a la estación de trabajo, donde el usuario procesa los datos en la forma en que éste desee.

Los sistemas basados en Servidores de Archivos tienen algunas limitaciones, una de éstas es que múltiples usuarios están imposibilitados de actualizar los mismos datos simultáneamente. Los archivos pueden ser accedidos sólo por una máquina en un tiempo.

Por ejemplo: si una estación de trabajo está procesando la consulta de un usuario acerca de " todos los clientes que viven en el Distrito Federal ", este requerimiento es enviado al servidor de archivos el cual a su vez enviará todo el archivo de clientes de vuelta a la estación de trabajo quien ejecutará la consulta y en la cual sólo aparecerán 5 clientes; esto en una red que cuente con un número mínimo de nodos no representará mayor problema, sin embargo es posible imaginar la degradación en tiempo de respuesta y el congestionamiento causado en redes con gran número de nodos.

A través de una Arquitectura cliente/servidor se pueden aprovechar y combinar las capacidades de equipos heterogéneos como una PC, una estación de trabajo, una minicomputadora o un mainframe, este esquema constituye la siguiente etapa de la computación basada en clientes.

La Arquitectura cliente/servidor es una forma de proceso distribuido, pero que toma el concepto del modelo basado en clientes y lo lleva más lejos. en este modelo un servidor es responsable solamente de repartir archivos y recursos de periféricos, así el servidor puede actuar como una computadora corriendo aplicaciones en red, ésto hace posible que en la Arquitectura cliente/servidor se distribuya el procesamiento de funciones entre las estaciones de trabajo y los servidores de una manera mucho más eficiente, así un servidor no sólo puede retornar archivos planos sino que puede llevar a cabo labores de ordenamiento, selección e indexación.

En esta arquitectura el servidor opera como un ente inteligente capaz de manejar los requerimientos de los clientes y sólo retornar los valores de las consultas hechas funcionando así como un servidor de bases de datos.

El desarrollo del servidor ha dado lugar a las siguientes ventajas:

- Capacidad para recuperar y almacenar grandes cantidades de datos
- Capacidad para asegurar la integridad del almacenamiento de datos
- La separación del almacenamiento físico y la presentación de datos, permitiendo a la base de datos ser actualizada sin impactar las aplicaciones que accesan a la base (independencia de datos)
- La capacidad para presentar múltiples vistas de datos como soporte de múltiples departamentos, aplicaciones y usuarios
- Soporte de administración de datos para controlar el acceso a los mismos
- Soporte de un lenguaje estándar SQL
- Capacidad para procesar cientos o tal vez miles de transacciones por segundo

# **CAPÍTULO III**

## **DISEÑO DE LA BASE DE DATOS**

## CAPÍTULO III DISEÑO DE LA BASE DE DATOS

### OBJETIVO DEL CAPÍTULO

Comprender las variantes que existen entre las bases de datos, así como las fases del diseño y selección de la metodología que cubra con los requisitos del sistema.

### INTRODUCCIÓN

El diseño de bases de datos se refiere al proceso de organizar los campos de datos necesarios para una o más aplicaciones, poniéndolos en una estructura organizada. Esta estructura debe contemplar las relaciones necesarias entre los campos cumpliendo al mismo tiempo con las restricciones físicas del sistema específico de manejo de bases de datos que se esté utilizando. Existen dos partes en el proceso: una es el diseño lógico de la base de datos y la otra es el diseño físico de la base de datos.

- El diseño lógico consta de dos componentes:

- a) El primero significa organizar los campos de datos en agrupamientos no redundantes basados en las relaciones de los datos.
- b) El segundo abarca una organización inicial de esos agrupamientos lógicos en estructuras basadas en la naturaleza del sistema del manejo de datos y de las aplicaciones que utilizarán los datos.

- El diseño físico depende de la implementación y toma los resultados del diseño lógico y lo refina de acuerdo a las características del sistema de manejo de datos que se este utilizando.

Varias razones hacen que el diseño cuidadoso de la base de datos sea esencial; estas incluyen la redundancia de datos, el desempeño en la aplicación, la independencia y seguridad de los datos y la facilidad de programación. Todos estos factores son importantes en el ambiente del procesamiento de datos, y todos pueden ser afectados cuando el diseño no es el apropiado.

#### Redundancia de Datos

El diseño adecuado de la base de datos es esencial para evitar las redundancias tanto de un archivo como entre archivos. La cantidad de redundancia aparente de los datos (campos llave y no llave) varía de acuerdo al sistema manejador de datos para el que se están diseñando los archivos. En todos los casos, al menos en las primeras etapas del diseño, el objetivo es el eliminar toda la redundancia entre los datos. En etapas posteriores al diseño,

conforme los datos toman forma de estructura del DBMS específico, puede o no existir redundancia abierta entre los campos.

### **Desempeño**

El desempeño, velocidad operacional de las aplicaciones y sistemas, puede verse afectado o influenciado por un cierto número de factores en el medio ambiente de la base de datos y que están bajo el control de quien la diseña. Muchos de éstos son específicos del producto que se está utilizando; esto es, depende de las características del DBMS particular, pero otros tienen una naturaleza lo suficientemente general que pueden eliminarse al hacer un diseño cuidadoso de la base de datos.

### **Independencia de los datos**

La independencia de los datos, o capacidad de modificar su estructura sin afectar programas ya existentes, es fundamentalmente una función del modelo de datos utilizado por el DBMS específico. Todos estos sistemas proporcionan un mejor grado de independencia de los datos al compararlos con el procesamiento normal de archivo. Más allá de los aspectos específicos de la estructura de los datos y del diseño de la base de datos, la forma en que se escribe un programa puede afectar la independencia de los datos.

### **Seguridad de los datos**

Por lo general los sistemas de manejo de la base de datos tienen una variedad de opciones interconstruidas para seguridad de los datos. Estas varían desde claves de acceso asociadas con usuario específico o con datos particulares, hasta varias formas de prohibir que todos excepto algunos usuarios efectúen ciertas operaciones con determinados datos. La seguridad de los datos se transforma en un tema del diseño de la base de datos cuando la cantidad mínima de datos que el sistema es capaz de devolver tomándolos de la base en una llamada de programa es mayor que la cantidad que la persona que ejecutó el programa tiene derecho a ver. En ese caso, es posible que la estructura tenga que diseñarse de manera que esos datos, que de otra forma estarían colocados juntos en una unidad recuperable, estén divididos en varias unidades.

### **Facilidad de programación**

Aunque el empleo de los sistemas de manejo de la base de datos requiere que los programadores aprendan nuevos conceptos y protocolos, el objetivo general del empleo de los DBMS es disminuir la complejidad de programación y las tasas de error de programación.

Dentro del nivel de programación mejorado que permite el empleo de los DBMS, algunas decisiones en el diseño de las bases de datos pueden en cierta medida dificultar o facilitar la tarea del programador. Desafortunadamente las decisiones que simplifican las especificaciones de programación tienen por lo general un efecto adverso sobre la redundancia de los datos, o en el desempeño del sistema o en ambos.

### **Análisis de los datos**

Antes de que se pueda comenzar la evaluación de los requerimientos de una base de datos, se deben de comprender los objetivos y ámbitos globales del sistema para el que se va a desarrollar la base de datos. Posteriormente se desarrolla un modelo de información completo y a detalle.

El modelo de información incluye un diccionario de datos que define todos los elementos de datos en términos de la información que se usa para desarrollar el elemento.

Posteriormente, se deben de definir las características lógicas y físicas de la base de datos. Con el modelo de la información y la especificación del modelo como guía, se define la organización lógica de los datos. La organización lógica debe considerar los requerimientos de acceso, modificación, asociación de los datos, así como otros aspectos orientados al sistema. Una vez que ha sido establecida la organización lógica de los datos, debe de desarrollarse la organización física. La organización física de una base de datos define la estructura de archivos, los formatos de los registros, las características de procesamientos dependientes del equipo y las características del DBMS. Finalmente, ha de realizarse una revisión completa del esquema y características físicas.

## **III.1 ETAPAS DEL PROCESO DE DISEÑO**

### **Elementos del Proceso de Diseño**

Las principales clases de entradas y resultados del proceso de diseño son:

#### Entradas:

- **Requerimientos de información general.**
- **Requerimientos de procesamiento**
- **Especificaciones de DBMS.**
- **Configuración del equipo/sistema operativo.**
- **Especificaciones del programa de aplicación.**

---

**Resultados:**

- Estructura lógica de la base de datos (vista del usuario)
- Estructura del diseño físico (almacenamiento)

Los requerimientos de información general representan las descripciones de varios usuarios de la organización para la cual los datos son reunidos, los objetivos de la base de datos, y las vistas de los usuarios de los cuales los datos deben de ser colectados y almacenados. Estos requerimientos son considerados como independientes de los procesos por que no están atados a ningún sistema de manejo de datos específico o aplicación. El diseño de base de datos basado en estos requerimientos es considerado ventajoso a largo plazo para las bases de datos que deben ser adaptables a los requerimientos cambiantes del procesamiento.

**Diseño de una Base de Datos**

El diseño de una base de datos, envuelve un proceso de tres etapas:

1. Definición de los datos.
2. Refinamiento de los datos.
3. Establecimiento de relaciones entre los atributos.

**Definición de los Datos**

En la primera etapa, en la definición de los datos se enlistan todos los atributos importantes involucrados en la aplicación. Para hacerlo, se debe de examinar la aplicación en detalle para determinar exactamente que tipo de información debe ser guardada en la base de datos.

Un punto importante a tener en cuenta durante esta etapa del diseño, es que se deben de listar todos los posibles atributos para el sistema de gestión de base de datos. Se pueden listar más atributos de los que realmente se necesitan para la aplicación en particular, no es ningún problema, ya que los atributos realmente innecesarios serán eliminados durante la etapa de refinamiento de datos.

**Refinamiento de Datos**

Durante esta etapa se refinan los atributos de la lista inicial, de forma que los atributos conformen una descripción exacta de los tipos de datos que se necesitarán en la



base de datos. En este punto del proceso es vital considerar sugerencias de tantos usuarios de la base de datos como sea posible.

En otros casos particulares, algunos refinamientos pueden hacerse evidentes rápidamente y otros no tanto. Así que, volviendo a repasar la lista de atributos se irán haciendo más obvios algunos refinamientos necesarios.

Mientras se refina la lista de atributos, es conveniente calcular la cantidad de espacio en caracteres que se necesitará para mantener la mayor entrada posible de cada atributo en concreto.

### **Establecimiento de las Relaciones**

Durante la tercera etapa, la representación de las relaciones entre los atributos puede ayudar a determinar cuales son importantes y cuales lo son menos. Una manera de determinar las relaciones entre los atributos es cuestionar las mismas preguntas que se le plantearán a la base de datos.

Cuando se establecen las relaciones entre variables, también se puede descubrir que se necesita un atributo adicional, mismo que se deberá añadir al diseño.

## **III.3 MODELO ENTIDAD-RELACIÓN**

### **Modelos de Datos**

Para definir la estructura de una base de datos es necesario definir el concepto de modelo de datos. Un modelo es una representación de objetos y eventos del mundo real. Un modelo de datos es una representación abstracta (una descripción) de los datos por medio de sus entidades, eventos, actividades y sus asociaciones dentro de una organización.

Los modelos lógicos se utilizan para describir los datos en los niveles conceptual y de vista. Son flexibles y permiten especificar claramente las limitantes de los datos.

El modelo de entidad-relación es una técnica de las más comunes para el diseño de bases de datos, y requiere de una serie de pasos para producir una estructura que sea aceptable para el DBMS. Es el representativo de la clase de los modelos lógicos basados en objetos.

### **Beneficios de uso de la metodología de Entidad-Relación (ER)**

Los enfoques convencionales para el diseño lógico de bases de datos normalmente sólo tienen una fase: convertir la información de los objetos del mundo real directamente al esquema del usuario. La metodología ER para el diseño lógico de bases de datos consiste en dos fases principales:

1. Definir el esquema empresarial usando diagramas entidad-relación y,
2. Traducir el esquema empresarial al esquema del usuario.

#### **Las ventajas son:**

1. La división de las funciones y el trabajo en dos fases hace que el diseño de la base de datos sea más simple y organizado.
2. El esquema empresarial es fácil de diseñar ya que no está restringido por las capacidades del DBMS y es independiente del almacenamiento y de consideraciones de eficiencia.
3. El esquema empresarial es más estable que el esquema del usuario. Si se desea cambiar de un DBMS a otro, el esquema empresarial será el mismo, ya que éste es independiente del DBMS que se use.
4. El esquema empresarial expresado por un diagrama entidad-relación es más fácil de entender por gente sin conocimientos de computación.

### **Conceptos en los que se basa la metodología de Entidad-Relación**

El modelo de datos ER se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones.

#### **Entidades**

Una entidad es una "cosa" que se puede distinguir. Las entidades se pueden clasificar en diferentes tipos de entidades; por ejemplo, empleado y departamento. En un diagrama entidad-relación (DER), las entidades se representan con un rectángulo. Un sustantivo corresponde al nombre de la entidad en el DER.

EMPLEADOS

DEPARTAMENTO

Una entidad es el objeto principal del cual se tiene que almacenar información, normalmente denotando una persona, lugar, cosa o evento de interés.

### Relaciones

Pueden existir relaciones (asociaciones) entre las entidades. Por ejemplo: trabaja-en, es una relación entre las entidades empleado y departamento.



En la notación de diagramas ER, una relación se representa con un rombo con líneas conectando las entidades relacionadas normalmente con un verbo correspondiente a la relación. También se debe de especificar el tipo de asociación de las relaciones (uno a uno, uno a muchos, muchos a muchos).

### III.3 DISEÑO DEL MODELO ENTIDAD-RELACION

Este diseño requiere varios pasos para producir una estructura que sea aceptable para el DBMS. Los pasos del diseño del modelo entidad-relación son:

- Análisis de datos.
- Producir y optimizar el modelo de entidad.
- Desarrollo del esquema lógico.
- Proceso del diseño de la base de datos física.

#### Fase de análisis de datos

En esta fase se identifican los recursos de datos de la organización. El enfoque de análisis de datos y el énfasis que se le da a cada tarea depende mucho de los objetivos del proyecto. El análisis de datos se usa para:

- Determinar las fuentes de datos fundamentales de la organización.
- Permitir el diseño de estructuras de archivos flexibles capaces de soportar varias aplicaciones relacionadas.
- Ayudar al desarrollo de la aplicación o conversión, proporcionando un entendimiento fundamental de los datos involucrados.

- Formar las bases para el control de los datos, seguridad y auditoria de las aplicaciones y sistemas resultantes.
- Organizar los hechos relevantes relacionados con la organización de los datos.
- Ayudar a la unificación de la organización indicando las partes comunes entre sus departamentos y los requerimientos de datos.
- Proporcionar las bases para evaluar la capacidad de estructuración de los sistemas de manejo de bases de datos.
- Identificar las entidades que son relevantes para solucionar el problema del procesamiento de datos.
- Determinar las relaciones entre estas entidades.
- Establecer definiciones de procesos y datos en un diccionario de datos.
- Producir el modelo de entidad.

El objetivo primario del análisis de datos es el de proporcionar las bases para el diseño, se deben de identificar las entidades que son necesarias para resolver el problema definido por el usuario. Durante las etapas iniciales del análisis, es posible que no se conozcan todos los atributos de todas las entidades. Sin embargo, a medida que éstos se determinen, se debe de documentar la definición del atributo y su papel en un diccionario de datos.

### Elaboración del Modelo de Entidad

Durante la fase de análisis se determinan las entidades mayores y sus relaciones. Estas entidades y sus relaciones se representan en Modelos de Entidad, el cual representa la relación entre las clases de entidades.

La representación nos permite incluir solo aquellas entidades que se requieren para resolver un problema particular del procesamiento de datos. El modelo de entidad es esencialmente una vista del mundo real de los datos.

Durante la fase del modelaje se definen las clases y relaciones de entidad más significativas, sin embargo el modelo deberá ser revisado, modificado o extendido como resultado del conocimiento sobre las nuevas entidades que se descubran. El modelo se usa para:

- Reducir la redundancia en las relaciones.
- Determinar cuáles entidades son significativas al modelo y a los requerimientos del usuario.
- Resolver las entidades no binarias entre entidades.

### **Etapas de la integración de los modelos de Entidad**

Las etapas requeridas para integrar los modelos de entidad son:

- Identificar **cada** sinónimo u homónimo en los diferentes modelos. Esta tarea es más fácil si **se usa un** diccionario de datos. Los componentes con homónimos deben ser **renombrados**. Los componentes con sinónimos deben de usar el mismo nombre.
- Los modelos de entidad para dos áreas de datos se integran suponiendo los tipos de **entidad que sean** idénticos o similares en los diferentes modelos de entidad. Esto **puede** incrementar el número total de atributos del tipo de entidad, ya que las **entidades** idénticas pueden usar diferentes atributos.
- Como **resultado** de la integración, el modelo compuesto de entidad puede contener **relaciones** redundantes. Esta redundancia **puede ser** eliminada, sin embargo, **determinar** las relaciones que son directamente significativas y cuáles son **redundantes puede** presentar dificultades que pueden ser resueltas solamente a través de un buen entendimiento del ambiente.

### **Combinación de modelos de Entidad**

**Al convertir bases de datos** existentes en su equivalente de modelo de entidad, el **diseñador puede llegar a diferentes** modelos dependiendo de los programas o aplicaciones de los cuales los **modelos** fueron derivados. Se debe de **intentar** remover las redundancias e inconsistencias al combinar los modelos de varios programas para quedarse con un modelo integrado. Esta combinación permitirá al diseñador determinar lo siguiente:

- Cuales son las clases de entidad y los atributos comunes.
- **Las** inconsistencias en los nombres y uso de los atributos. Estas inconsistencias existen cuando **dos** entidades con **diferentes** nombres se ve que son la misma entidad.
- La eficacia del modelo en términos de satisfacer las necesidades del usuario.

- Si algunos atributos considerados en una entidad son realmente miembros de otra clase de entidad o de nuevas clases de entidad.
- La existencia de inconsistencias en las relaciones.

Este modelo combinado puede ser usado como la estructura para revisiones posteriores para llegar a un modelo de entidad integrado que sirva a un área de datos mayor, en lugar de varios modelos pequeños orientados a unas cuantas aplicaciones.

### **Agrupamiento de clases de Entidades**

El agrupamiento de clases de entidades puede hacerse en la etapa de diseño físico o lógico. En la etapa del diseño físico esto puede hacerse basándose en consideraciones de desempeño. Las clases de entidades pueden juntarse o separarse en diferentes bases físicas dependiendo de los requerimientos de acceso.

El agrupamiento lógico de clases de entidades depende de la naturaleza de los datos y sus estructuras. Es necesario, pero no suficiente, decir que los atributos se agrupan en una clase de entidad, y éstas se agrupan en un modelo de entidad para satisfacer los requerimientos de procesamiento. El agrupamiento de clases de entidad se hace enteramente para satisfacer los siguientes requisitos:

- El área a que sirven los datos o de la cual se originan.
- La estructura de datos inherente.
- La vista del usuario.
- Los usos de los datos.
- Las consultas que se hacen de los datos.
- Las necesidades de procesamiento del usuario.

## **III.4 DISEÑO DEL ESQUEMA LÓGICO**

La vista de la aplicación puede definirse como el conjunto de datos que son requeridos por una aplicación particular para satisfacer una necesidad específica de procesamiento de datos. Tenemos vistas de aplicación de:

- Una clase de entidad.
- Agrupamientos de clases de entidades.
- Agrupamientos de clases de entidad y base de datos físicas
- Agrupamiento de bases de datos físicas.

El esquema lógico puede ser definido como el mapeo del modelo de entidad en la construcción proporcionada por el manejador de la base de datos. En general el esquema lógico indica como se almacenará y accederá el modelo. En el diseño del esquema lógico tal vez sea necesario hacer algunos cambios al modelo para adecuarse al DBMS. El modelo de entidad no es el esquema lógico. El modelo de entidad tiene las siguientes características:

- Es una representación de la vista de datos del mundo real.
- Proporciona las bases para continuar con el análisis y diseño de la base de datos.
- No está restringido a ningún sistema manejador de bases de datos (DBMS).
- No es implementable directamente.
- Una estructura estable de referencia a la cual se puede agregar nuevas entidades, atributos y relaciones si la organización así lo quiere.

### **III.5 CONVERSIÓN DEL ESQUEMA LÓGICO A UNA BASE DE DATOS FÍSICA**

Los detalles de esta fase dependen de las características del manejador escogido para el diseño. Esta transformación requiere las siguientes selecciones:

- Bases de datos físicas y tipos de relaciones lógicas, ya sea unidireccionales o bidireccionales, relacionadas físicamente.
- Métodos de acceso.
- Segmentos, estructuras jerárquicas y representaciones de datos, incluyendo tipo y tamaño.
- Índices secundarios.
- Tipos de apuntadores en la relación.

Adicionalmente de las selecciones anteriores, la implementación también incluye:

- Asignación de dispositivos de almacenamiento.
- Carga y organización de las bases de datos.

El esquema lógico debe hacerse de manera que lo único que se deje a los diseñadores de la base física sea la selección de los métodos de acceso y los índices secundarios. Hubard indica que se deben de seguir las siguientes reglas durante el diseño físico:

- Cada clase de entidad debe ser tratada como una base de datos física.
- Si dos clases de entidades comparten una relación entre un atributo y la llave primaria por lo menos, entonces las estructuras deben de consistir en dos bases de datos físicas con conexión virtual o física entre ellas.
- Las relaciones padre-hijo deben ser definidas en una sola base de datos física.
- Los segmentos que se usen frecuentemente deben ser mantenidos lo más cercanamente posible a su raíz.
- Reducir el tiempo de búsqueda de grupos de datos grandes usando índices secundarios.
- Los segmentos de tamaños variables no deben ser colocados en el mismo grupo de datos si se hacen inserciones y borrados frecuentes.

### III.6 SISTEMAS MANEJADORES DE BASES DE DATOS

Sistema Manejador de Bases de Datos cuyas siglas en inglés son DBMS (Data Base Management System). Posee una estructura sencilla compuesta por dos partes: una Sección Posterior y un conjunto de Secciones Frontales.

La sección posterior permite llevar a cabo todas las funciones básicas de un DBMS, como son: la definición de datos, manipulación de los datos, seguridad, integridad, etc.

Por el otro lado, las secciones frontales son las diversas aplicaciones ejecutadas dentro del DBMS, tanto las escritas por los usuarios como las "integradas" que son las proporcionadas por el proveedor del DBMS o bien por otros proveedores de programas, no habiendo diferencia alguna entre ambas para la sección posterior.

En otras palabras, entre la base de datos física misma y los usuarios del sistema existe un nivel de programas conocidos comúnmente como el Sistema Manejador de Bases de



Datos (DBMS). El DBMS maneja todas las solicitudes de acceso formuladas por los usuarios a la base de datos. Así una de las funciones generales de DBMS es distanciar a los usuarios de la base de datos a nivel de equipo de manera muy similar a la forma como los sistemas de lenguajes de programación evitan a los programadores de aplicaciones la necesidad de ocuparse de detalles al nivel de lenguaje de máquina. De esta forma, el DBMS presenta a los usuarios una vista de la base de datos en un nivel un tanto por encima del nivel del equipo y hace posibles sus operaciones expresadas en términos de esa vista de nivel más alto.

El sistema manejador de la base de datos es por supuesto el conjunto de programas que maneja todo el acceso a la base de datos. Conceptualmente lo que sucede es lo siguiente:

1. Un usuario solicita acceso, empleando algún sublenguaje de datos determinado, por ejemplo SQL.
2. El DBMS interpreta esa solicitud y la analiza.
3. El DBMS inspecciona en orden, el esquema externo de ese usuario, la correspondencia externa/conceptual asociada, el esquema conceptual, la correspondencia conceptual/interno, y la definición de la estructura de almacenamiento.
4. El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenada.

La descripción anterior hace pensar que todo el proceso es interpretativo, pues sugiere que los procesos de analizar la solicitud, inspeccionar los diversos esquemas, etc., se realizan todos al momento de la ejecución. La interpretación, por su parte, casi siempre implica un desempeño pobre debido al aumento de tiempo de ejecución. Sin embargo, en la práctica puede ser posible compilar las solicitudes de acceso antes del momento de la ejecución.

Examinando las funciones del DBMS con un poco más de detalle, encontramos que dentro de dichas funciones se incluyen por lo menos las siguientes:

### **Definición de los datos**

El DBMS debe ser capaz de aceptar definiciones de datos (esquemas externos, el esquema conceptual, el esquema interno, y todas las correspondencias asociadas) en versión fuente y convertirlas en la versión objeto apropiada. Dicho de otro modo, el DBMS debe incluir componentes procesadores de lenguaje para cada uno de los diversos Lenguajes de Definición de Datos (por sus siglas en inglés DDL, Data Definition Language). El DBMS también debe entender las definiciones del DDL y debe poder utilizar estos conocimientos para interpretar y responder las solicitudes de los usuarios.

### **Manipulación de los datos**

El DBMS debe ser capaz de atender las solicitudes del usuario para extraer, y quizá poner al día datos que ya existen en la base de datos o para agregar en ella datos nuevos; en otras palabras, el DBMS debe incluir un componente procesador de Lenguaje de Manipulación de Datos (por sus siglas en inglés DML, Data Manipulation Language).

En general, las solicitudes en el DML pueden ser "planeadas" o "no planeadas".

Una solicitud planeada es aquella cuya necesidad se previó mucho tiempo antes de que tuviera que ejecutarse por primera vez. El administrador de la base de datos (por sus siglas en inglés DBA, Data Base Administrator) habrá afinado con toda probabilidad el diseño físico de la base de datos a fin de garantizar un buen desempeño para estas solicitudes.

Una solicitud no planeada, en cambio, es aquella cuya necesidad no se previó sino que surgió de improviso. El diseño de la base de datos puede ser o no ideal para la solicitud específica de que se trate. En general el logro del mejor desempeño posible con solicitudes no planeadas representan un reto considerable para el DBMS.

Las solicitudes planeadas son características de las aplicaciones "operacionales" o de "producción"; las no planeadas son representativas de las aplicaciones de "apoyo a decisiones". De esta forma, las solicitudes planeadas casi siempre se originan en programas de aplicación previamente escritos, en tanto que las solicitudes no planeadas, por definición, se emitirán de manera interactiva.

### **Seguridad e integridad de los datos**

El DBMS debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el administrador de la base de datos.

### **Recuperación y concurrencia de los datos**

El DBMS o en su defecto algún componente de programas relacionado con él, al que por lo regular se denomina administrador de transacciones, debe cuidar del cumplimiento de ciertos controles de recuperación y concurrencia.

### **Diccionario de datos**

El DBMS debe incluir una función de Diccionario de datos. Puede decirse que el Diccionario de datos es una base de datos por derecho propio, pero una base de datos del sistema, no del usuario. El contenido del diccionario puede considerarse como datos acerca

de los datos, es decir, definiciones de otros objetos en el sistema, y no sólo de datos en bruto. En particular, en el diccionario de datos se almacenarán físicamente todos los diversos esquemas y correspondencias tanto de sus versiones fuente como en las versiones objeto. Un diccionario completo incluirá también referencias cruzadas para iniciar cuáles programas usan cuáles partes de la base de datos, cuáles usuarios requieren cuáles informes, qué terminales están conectadas al sistema, y cosas por el estilo. Es más, el diccionario podría estar integrado a la base de datos a la cual define, e incluir por tanto su propia definición. Deberá ser posible consultar el diccionario igual que cualquier otra base de datos de modo que se pueda saber cuáles programas o usuarios podrían verse afectados por alguna modificación propuesta para el sistema.

### Desempeño

Por último, el DBMS deberá ejecutar todas las funciones antes identificadas en la forma más eficiente posible.

Como conclusión, en forma de resumir todo lo anterior se puede decir que el DBMS constituye la *interfaz* entre el usuario y el sistema de la base de datos. La interfaz del usuario puede definirse como una frontera del sistema, más allá de la cual todo resulta invisible para el usuario.

El DBMS es definitivamente el componente de programas más importante de todo el sistema, pero no es el único. Entre los demás puede mencionarse las utilerías las herramientas para desarrollar aplicaciones, las ayudas para el diseño, los generadores de información, etc. También es importante hacer notar que los DBMS's se clasifican de acuerdo al tipo de base de datos que manejan. (Jerárquicas, Reticulares o Relacionales).

Una vez definido lo que es un sistema manejador de bases de datos y sus funciones principales, pasaremos a describir tres de los más importantes DBMS's que ofrece el mercado hoy en día a los usuarios de sistemas de bases de datos. El tipo de DBMS que describimos es el que maneja bases de datos relacionales RDBMS (Relational Data Base Management System), ya que es el modelo de base de datos más poderoso y el más usado actualmente.

### III.6.1 Oracle7

La base de datos ORACLE es ampliamente usada y ocupa una posición importante en el ambiente de las bases de datos sobre el cual tiene una gran influencia. Junto con INGRES es pionera en la implementación de bases de datos relacionales, y encabeza el mercado en la promoción de programas portable. Se puede decir que fue un de las primeras en adoptar al SQL (Structured Query Language) como un lenguaje estándar de acceso.

De ningún modo se puede decir que la portabilidad es el único aspecto importante de ORACLE ya que tiene la habilidad para trabajar en las más populares plataformas de equipo y explotar los diferentes ambientes tales como es el caso del equipo agrupado (VAX y RS/6000), máquinas SPM, y masivamente equipo paralelo. Para ofrecer esto, ORACLE ha desarrollado una arquitectura de programas flexible y escalable, la cual varía necesariamente entre cada máquina.

La versión 7 de ORACLE es un Sistema Manejador de Bases de Datos Relacional, escalable y completamente portable a más de 80 plataformas de equipo y sistemas operativos, desde sistemas de escritorio a computadoras grandes y supercomputadoras. Este RDBMS es adaptable a una arquitectura de Servidor Multihebras (Multi-Threaded), rindiendo un alto desempeño escalable a un gran número de usuarios en todas las arquitecturas de equipo, incluyendo multiprocesos simétricos, ORACLE7 ofrece control de concurrencia con bloqueo completo y no restringido a nivel de registro, además de contención libre de consultas. Las aplicaciones de ORACLE7 pueden correr igual que el reloj, debido a que las funciones del sistema tales como: respaldo, recuperación y la administración de la base de datos ocurren en línea sin interrumpir la transacción del procesamiento. Con este RDBMS, tanto los desarrolladores como los usuarios finales podrán tratar una base de datos distribuida física como una sola base de datos lógica.

Mientras los diferentes productos competidores ofrecen bastantes capacidades distribuidas avanzadas, ORACLE7 ofrece poco más que accesos remotos de lectura y escritura. Otros productos proporcionan funcionalidad de bases de datos en forma de iniciadores, procedimientos almacenados, reglas de integridad, etc., pero ORACLE7 ofrece un poco de cada uno de estos aspectos.

ORACLE7 provee su propio lenguaje SQL y los procedimientos de acceso a los datos desde al menos una fuente con SQL. Una sola sentencia SQL puede consultar datos desde múltiples bases de datos y realizar complejas uniones de tablas (Joins). Como se mencionó anteriormente, corre en la mayoría de las plataformas de equipo y sobre más de 40 sistemas operativos tales como: UNIX, VMS, MVS, VM, HP MPE/XL, Siemens, ICL, OS/2, Macintosh y Novell Netware.

Debido a lo extenso de su portabilidad, la base de datos ORACLE7 se encuentra por sí misma en competencia con una amplia variedad de productos que van desde DB2 de IBM en las mainframes hasta Gupta's SQL Base en una PC LAN. (Personal Computer Local Area Network). Existe una variedad de plataformas tal como Prime, Data General o la

Apple Mac donde ORACLE7 es la única opción real disponible de bases de datos portable, y en varias de estas plataformas a significado un porcentaje importante del mercado.

### Ventajas

La principal ventaja que ofrece ORACLE7 sobre sus principales competidores, es que es un sistema escalable y completamente portable a más de 80 plataformas de equipo, que van desde sistemas de escritorio hasta computadoras grandes y supercomputadoras, y obviamente corriendo bajo una amplia variedad de sistemas operativos.

Detalles del Producto	Descripción
Vendedor	Oracle
Fecha de la primera versión del producto	1979
Última versión y fecha de la liberación	7.0 (Julio 1992)
Fecha estimada para la próxima versión	Por Anunciar

Plataformas Equipo	Soportado
Equipo basado en UNIX	SI
Mainframes IBM	SI
Digital VAX	SI
IBM PC	SI
Otros	ICL, VME

Sistemas Operativos	Soportado
UNIX	SI
MVS	NO
VM	NO
VMS	SI
OS/2	SI
MS-DOS	Planeado
Otros Sistemas Operativos	DG/AOS, BTOS, Primos, GCOS, VOS, HP, MPE/XC
Programas de comunicación	TCP/IP, SPX/IPX, LU 6.2, etc.

Tablas III.6.1.1 Detalles, Equipo y Sistema Operativo ORACLE

Parámetros Básicos del Producto	Soportado
Número máximo de registros por tabla	Depende del Equipo
Número máximo de columnas por registro	Depende del Equipo
Número máximo de caracteres por campo alfanumérico	Depende del Equipo
Número máximo de tablas por base de datos	Depende del Equipo
Memoria Total mínima recomendada (en el servidor)	Depende del Equipo
Memoria recomendada por usuario (en el servidor)	Depende del Equipo
Memoria recomendada por usuario (en el servidor)	Depende del Equipo
Espacio en disco mínimo requerido	Depende del Equipo
Número máximo de usuarios por base de datos	Depende del Equipo

Tipos de Datos	Soportado
Booleano	NO
Punto flotante	NO
Punto Fijo ó Money	NO
Caracter de longitud variable	SI
Fecha	SI
Hora	SI
Texto Largo	VARChar - 52K Largo - 52GB
Objetos grandes (IMAGENES)	Long RAW - 52GB
Tipos de datos definidos por el usuario	NO
Tipos de datos compuestos definidos por el usuario	NO
Especificación de reglas de conversión	NO
Distinción entre el valor NULL y los valores del dominio	SI

Tablas III.6.1.2 Parámetros básicos y tipos de datos ORACLE

En cuanto a aspectos de funcionalidad se refiere, ORACLE7 ofrece iniciadores, procedimientos almacenados, alertas de las bases de datos entre otras. También incluye capacidades distribuidas para el manejo de bases de datos distribuidas, con la cual los usuarios finales podrán tratar varias bases de datos físicas distribuida como una sola base de datos lógica.

ORACLE7 también proporciona herramientas que permite a los usuarios construir compuertas SQL para otras bases de datos y sistemas de archivos. Esta compuerta programable soporta llamadas de procedimientos remotos para todo tipo de sistemas externos.

Finalmente, ORACLE7 proporciona un conjunto de herramientas de desarrollo de aplicaciones muy importantes, entre las cuales se encuentran el SQL\*Forms y SQL\*Plus, que son las principales, pero otros componentes tratan con la generación de reportes, manejo de redes y facilidades de usuario final, tales como las hojas de cálculo y los gráficos, todo esto gracias a la utilización de lenguajes de cuarta generación (4GL).

### **Desventajas**

Se podría decir que la principal desventaja que presenta ORACLE7, es su gran demanda de memoria para poder almacenar datos en ella, y si la memoria es insuficiente, es probable que el disco ocasione un cuello de botella.

Otra desventaja que presenta ORACLE7 con respecto a sus competidores, es la de que todavía no ofrece un ambiente significativo de Programación Orientada a Objetos, además de que no explota completamente las plataformas paralelas.

### **III.6.2 Ingres**

La base de datos INGRES lleva la batuta y aventaja en algunos aspectos a la mayoría de sus competidores al proporcionar funcionalidad distribuida, optimización de consultas sofisticadas, optimización distribuida, manejo de objetos y alertas de la base de datos.

Actualmente, el énfasis de los productos INGRES está orientado fuertemente hacia los objetos. Para esto, INGRES provee uno de los pocos lenguajes de cuarta generación (4GL) orientado a objetos disponibles en el mercado y posee capacidades para el manejo de objetos dentro de la base de datos, lo que lo hace estar por encima de sus competidores. La popularidad de Microsoft Windows ha promovido un fuerte interés en IngresWindows 4GL, pero el interés en las capacidades de objetos de la base de datos INGRES ha sido menor, tal vez por que los usuarios no han logrado comprender el potencial de sus usos y beneficios.

Para ofrecer soporte a sistemas de producción y ambientes OLTP, el manejador de bases de datos realacional, INGRES, permite un modelo de organización de los negocios a través del manejo de datos, el conocimiento, y los objetos. Está diseñado para proveer manejo de los datos y capacidades para desarrollar aplicaciones a través de diferentes plataformas de equipo y programas.

INGRES proporciona la habilidad para definir múltiples Servidores Multi-Hebra para acceder y actualizar datos compartidos. El manejo del conocimiento facilita esfuerzos de políticas en las empresas y asegura la integridad referencial. INGRES Event Alert puede usarse para crear aplicaciones que respondan dinámicamente a los requerimientos de la empresa. Los objetos pueden ser definidos y manejados vía los comandos SQL.

Los módulos de INGRES como son: INGRES/Windows4GL, un lenguaje de cuarta generación (4GL) orientado a objetos y manejador de aplicaciones, y el INGRES/Visión, un desarrollador de aplicaciones están disponibles por separado. Las aplicaciones corren idénticamente en cualquier sistema y los datos pueden procesarse desde cualquier sitio. INGRES es portable en IBM, DEC, Data General, HP, ICL, Pyramid, Sequent, UNISYS y plataformas DOS, y en sistemas corriendo bajo los sistemas operativos: VMS, ULTRIX, MVS, VM/CMS, UNIX y DOS. La arquitectura del servidor INGRES provee procesamiento tanto para el servidor como para los CPUs anfitriones.

En las tablas se pueden observar algunos datos técnicos de interés característicos de INGRES proporcionados por su fabricante.

### Ventajas

Al igual que la mayoría de RDBMS, INGRES está diseñado para proporcionar un manejo eficiente de los datos, además de ofrecer capacidades para que los usuarios puedan desarrollar aplicaciones bajo diferentes plataformas de equipo y programas. Así, para los usuarios resulta transparente el hecho de que los datos puedan ser procesados desde cualquier sitio y las aplicaciones se ejecuten idénticamente en los diferentes sistemas.

Otros aspectos en los que INGRES aventaja a la competencia, es en cuanto a su excelente funcionalidad y optimización distribuidas, por supuesto el manejo de objetos y las alertas de bases de datos.

El énfasis de los productos INGRES está orientado fuertemente hacia los objetos, por lo que provee uno de los pocos lenguajes de cuarta generación orientado a los objetos disponibles en el mercado.

Provee soporte a redes a través de sus módulos Ingres Net e Ingres Star, los cuales ofrecen capacidades de cliente/servidor por medio de diferentes ambientes de red y el manejo de bases de datos distribuidas permitiendo la existencia de un diccionario de datos global.



Finalmente, INGRES ofrecen un conjunto de herramientas que conforman un ambiente integrado de desarrollo bastante completo.

Detalles del Producto	Descripción
Vendedor	Ingres
Fecha de la primera versión del producto	1980
Última versión y fecha de liberación	6.4 (Nov. 1991)
Fecha estimada para la próxima versión	6.5 (1993)

Plataformas Equipo	Soportado
Equipo basado en UNIX	SI
Mainframes IBM	SI
Digital VAX	SI
IBM PC	SI
Otros	Bull & ICL

Sistemas Operativos	Soportado
UNIX	SI
MVS	SI
VM	NO
VMS	SI
OS/2	SI
MS-DOS	SI
Otros Sistema Operativos	VME,Bull, VMS de seguridad
Programas de comunicación	TCP/IP, SPX/IPX, AT&T

**Tabla III.6.2.1 Detalles, equipo y sistema operativo INGRES**

Parámetros Básicos del Producto	Soportado
Número máximo de registros por tabla	No limitado
Número máximo de columnas por registro	300
Número máximo de caracteres por campo alfanumérico	2008
Número máximo de tablas por base de datos	No limitado
Memoria Total mínima recomendada (en el servidor)	8 MB
Memoria recomendada por usuario (en el servidor)	50 K - 150 K
Espacio en disco mínimo requerido	80 MB Servidor
Número máximo de usuarios por base de datos	No limitado

Tipos de Datos	Soportado
Booleano	SI
Punto flotante	SI
Punto fijo	SI
Carácter de longitud variable	SI
Fecha	SI
Hora	SI
Texto largo	SI
Objetos grandes (IMAGENES)	SI
Tipos de datos definidos por el usuario	SI
Tipos de datos compuestos definidos por el usuario	NO
Especificación de reglas de conversión	SI
Distinción entre el valor NULL y los valores del dominio	SI

**Tablas III.6.2.2 Parámetros básicos y tipos de datos INGRES**

### Desventajas

Uno de los aspectos en los que hay que tener mucho cuidado, es en la implementación del diseño de la base de datos INGRES, ya que de lo contrario estará propensa a los cuellos de botella durante la consulta y actualización en sistemas de uso intensivo.

Otra desventaja es el soporte de la lectura sucia (Dirty Read), la cual permite a un usuario leer datos sin considerar a los demás usuarios que están realizando operaciones de inserción, actualización y borrado, lo que no garantiza que las vistas de la base de datos sean consistentes.

### III.6.3 Sybase

Es difícil establecer la filosofía para un producto como SYBASE. Pero ciertamente su facilidad de uso es relevante para el trabajo de los administradores de bases de datos. El énfasis está en decidir qué hacer sin tener la preocupación de como hacerlo. En varios aspectos esto es una excelente aproximación, pero su desempeño es bajo, ya que las funciones automatizadas probablemente nunca igualarán a los ajustes en línea específicos de un ambiente en particular. Un mayor énfasis está en la conectividad con otros ambientes, un área donde SYBASE ha tenido liderazgo sobre muchos productos de la competencia.

En la próxima versión de SYBASE las tres características más importantes serán: las herramientas orientadas a objetos, alta capacidad de redes distribuidas y soporte para sistemas grandes. La versión más reciente incluye un desarrollo conjunto con NCR.

SYBASE es un sistema manejador de bases de datos relacional con la capacidad de ejecutar aplicaciones en línea. Este RDBMS está basado en la arquitectura cliente/servidor y el Servidor SQL, en el cual las funciones de manipulación de los datos pueden ser manejadas separadamente desde las funciones de interfaz de usuario.

El Servidor SQL de Sybase maneja los datos y la memoria, teniendo la capacidad de administrar múltiples bases de datos y múltiples usuarios. Registra la localización actual de los datos en el disco, mapeando la descripción lógica de los datos con sus datos físicos almacenados. Mantiene datos y procedimientos caches siempre en la memoria. El Servidor SQL compila y ejecuta sentencias T-SQL (Transact SQL), siempre y cuando las peticiones sean hechas en un formato estándar de SQL, retomando siempre los resultados de los programas cliente.

El Transact-SQL que maneja el servidor SQL de Sybase es una versión mejorada del ANSI Standard. Este incluye construcciones para : 1) Definición de Datos; 2) Manipulación, y 3) Control de Datos. Dentro de las mejoras de Sybase se incluyen las siguientes:

- Control de Flujo se Sentencias.
- Tipos de datos definidos por el usuario y otros tipos de datos adicionales.
- Procedimientos almacenados y Iniciadores.

Al igual que ORACLE7 e INGRES, el Servidor SQL SYBASE está diseñado bajo una arquitectura Multi-hebra o Multi-hilo, en donde el servidor por sí mismo puede administrar a múltiples usuarios. No se apoya en el sistema operativo de la computadora central (host) para llevar a cabo la multi-tarea, reduciéndole de esta forma la carga de trabajo.

El servidor SQL de SYBASE está estructurado para que pueda trabajar en ambientes de RED, no siendo un producto de máquina-única modificado para red. El diseño del servidor reduce significativamente el tráfico de la RED. Los Clientes y Servidores pueden estar físicamente separados en máquinas diferentes. El acceso se hace especificando un Nombre Lógico el cual representa al Servidor; este nombre es trasladado a una dirección de la red via un archivo ASCII pequeño el cual reside en el ambiente de cada máquina (Archivo de Interfaz). El mismo mecanismo también es usado para habilitar que múltiples residan en una misma máquina y así permitir que los servidores se comuniquen con otros servidores.

SYBASE proporciona un conjunto de herramientas SQL (SQL Toolset) que tienen una variedad de aplicaciones. Dentro de estas herramientas se encuentran las siguientes:

- ISQL 8SQL Interactivo para consultas al momento
- Data WorkBench para trabajar con SQL y Report WorkBench
- Bulk Copy para transferencias dentro y fuera de la base de datos
- APT para el desarrollo de aplicaciones con ventanas
- Etc.

El Servidor SQL está disponible para un amplio rango de sistemas de computación que van desde PC's hasta Minicomputadoras.

En las tablas III.6.3.1 y III.6.3.2 se pueden observar algunos datos técnicos de interés proporcionados por el fabricante de SYBASE.

### Ventajas

Dentro de las ventajas que ofrece SYBASE está su arquitectura de cliente/servidor y el Servidor SQL que permiten la manipulación de datos separadamente desde las interfaces de usuario.

La arquitectura cliente/servidor ofrece una excelente conectividad con otras plataformas de equipo y programas.

El Servidor SQL está estructurado para soportar ambientes de redes y su diseño reduce significativamente el tráfico en éstas.

Detalles del Producto	Descripción
Vendedor	Sybase Software
Fecha de la primera versión y fecha de liberación	1987
Última versión y fecha de liberación	4.8 (1992)
Fecha estimada para la próxima versión	5.0 (Mediados 1993)

Plataformas Equipo	Soportado
Equipo basado en UNIX	SI
Mainframes IBM	SI
Digital VAX	SI
IBM PC	SI
Otros	Stratus

Sistemas Operativos	Soportado
UNIX	SI
MVS	SI
VM	NO
VMS	SI
OS/2	SI
MS-DOS	SI
Otros S.O.	
Programas de Comunicación	TCP/IP, Novell, DECNET, SNA, LU 6.2 & LAN Manager

**Tabla III.6.3.1 Detalles, Equipo y Sistema Operativo SYBASE**

Parámetros Básicos del Producto	Soportado
Número máximo de registros por tabla	No limitado
Número máximo de columnas por registro	255
Número máximo de caracteres por campo alfanumérico	2 GB
Número máximo de tablas por base de datos	32,000
Memoria Total mínima recomendada (en el servidor)	8 MB
Memoria recomendada por usuario (en el servidor)	40 KB
Espacio en disco mínimo requerido	50 MB Servidor
Número máximo de usuarios por base de datos	Sobre 1024

Tipos de Datos	Soportado
Booleano	SI
Punto flotante	SI
Punto fijo	SI
Carácter de longitud variable	SI
Fecha	SI
Hora	SI
Texto largo	SI
Objetos grandes (IMAGENES)	SI
Tipos de datos definidos por el usuario	SI
Tipos de datos compuestos definidos por el usuario	NO
Especificación de reglas de conversión	SI
Distinción entre el valor NULL y los valores del dominio	SI

**Tablas III.6.3.2 Parámetros Básicos y Tipos de Datos SYBASE**

### **Desventajas**

Existen dudas con respecto al desempeño de SYBASE en ambientes grandes OLTP, ya que no ofrece el soporte requerido por los programadores en estos ambientes.

Una de las desventajas más significativas es la falta de portabilidad total, ya que las aplicaciones desarrolladas en una plataforma no pueden ser fácilmente ejecutadas bajo otra diferente, esto ya se tomo en cuenta para la versión SQL Server 11.





# **CAPÍTULO IV**

## ***ELECCIÓN DEL DBMS Y LA INTERFAZ AL USUARIO***



## **CAPÍTULO IV**

### **ELECCIÓN DEL DBMS Y LA INTERFAZ AL USUARIO**

#### **OBJETIVO DEL CAPÍTULO**

Elegir el Manejador de Bases de Datos que mejor cumpla los criterios de evaluación propuestos para el Sistema y posteriormente seleccionar una herramienta de desarrollo que se adapte satisfactoriamente a los requerimientos de la DGENP y las ENPs.

#### **IV.1 ELECCIÓN DEL MANEJADOR DE BASES DE DATOS**

Antes de llevar a cabo el desarrollo y la implementación del nuevo sistema, se analizaron diferentes Manejadores de Bases de Datos, requerimientos del sistema para el manejo de información, herramientas para programación y facilidad de uso.

Para la elección del DBMS, se partió de la forma en como debía de interactuar con el usuario, la manera en que a éste le es más común pensar respecto a la organización de sus datos, por lo tanto, la consideración por preferencia del usuario es manejar formas tabulares para organizarlos, ya que representa mayor sencillez de comprensión y manejo de la información. Además, esto no sólo es significativo para el usuario, sino también para las personas involucradas en el diseño y programación, por la facilidad que implica relacionar distintas tablas.

Por tales motivos, se pensó en DBMS's que manejaran la filosofía del modelo relacional de datos y el poder que éste le proporciona a las computadoras en cuanto a la rapidez, capacidad de almacenamiento, aplicaciones portables, conectividad y manejo.

##### **IV.1.1 Criterios de evaluación**

Los productos que se analizaron, cumplan con la filosofía del Modelo Relacional de Datos y con los requerimientos del sistema.

Las herramientas analizadas fueron:

- ORACLE
- INGRES
- SYBASE

Para llevar a cabo la selección, se consideraron los siguientes criterios generales:

- Estructura y Arquitectura
- Funcionalidad general
- Funcionalidad distribuida
- Desempeño
- Manejo de CPU
- Memoria
- Entrada/Salida Disco
- Ambiente de desarrollo
- Utilerías

Para cada uno de los criterios se analizaron los tres DBMS elegidos para poder llegar a la elección de uno.

#### **IV.1.1.1 Estructura y Arquitectura**

##### **ORACLE**

Con la versión 7, la base de datos básica de ORACLE incluye funciones de rendimiento, tales como el limitador de recursos y las herramientas del Administrador de la Base de Datos (DBA, Data Base Administrator).

Son tres las opciones que caracterizan la estructura de la base de datos ORACLE; una opción Procedimental que provee iniciadores (iniciadores), procedimientos almacenados, alertas de la base de datos y otras funciones, una opción Distribuida que incluye todas las capacidades distribuidas valga la redundancia, que han sido añadidas, y por último una opción de Servidor Paralelo, la cuál fué diseñada para combatir la pérdida masiva de información por medio de plataformas paralelas. En adición a lo anterior, ORACLE también puede proporcionar una compuerta transparente de herramientas (Transparent Gateway) que permite a los usuarios construir compuertas SQL para otras bases de datos y sistemas de archivos. Esta compuerta programable soporta llamadas de procedimientos remotos para todo tiempo de sistemas externos.

La arquitectura interna de la versión 7 se muestra en la figura IV.1.1.1.1. En donde los almacenamientos temporales, colas, procedimientos SQL compartidos y los accesos repetidos, están contenidos dentro del "Área Global del Sistema" (SGA; System Global Area). Los usuarios pueden estar configurados para tener un Servidor Dedicado o un Servidor Compartido.

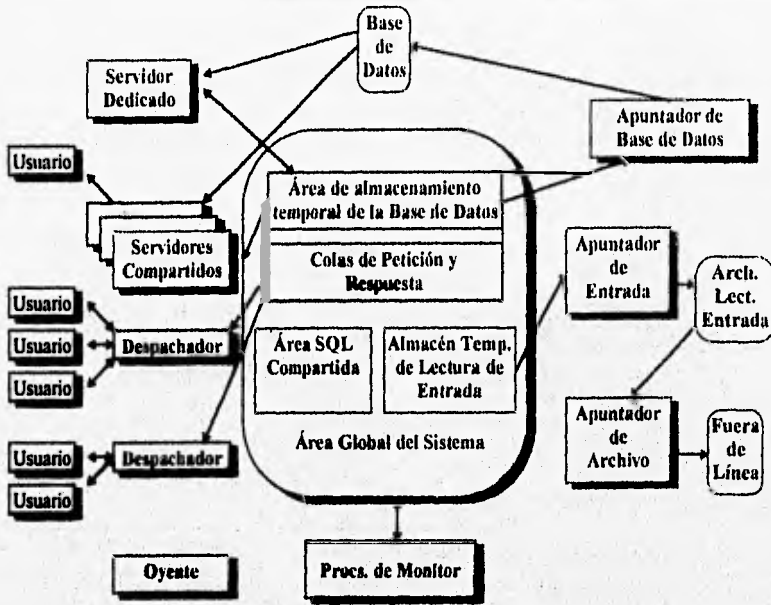


Figura IV.1.1.1.1 Arquitectura interna de ORACLE 7

## INGRES

El conjunto de productos INGRES es extenso, éste incluye una selección de lenguajes de cuarta generación (4GL's) y una variedad de otras opciones de desarrollo incluyendo un generador de código. El soporte a redes es proporcionado por los módulos INGRES Net e INGRES Star. INGRES Net ofrece las capacidades de cliente/servidor a través de una variedad de ambientes de red tales como DESCNet, TCP/IP, SNA, OSI, Novell SPX/IPX y otros. Por otro lado, INGRES Star provee el manejo de bases de datos distribuidas, permitiendo la implementación de un diccionario de datos global. En adición a esto, INGRES Gateways provee acceso a SQL de otras bases de datos y archivos, incluyendo DB2, SQL/DS, IMS y RDB.

Con lo que respecta a las capacidades relacionales de INGRES, este incluye características de seguridad y de puesta a tono, tal como es el Manejador de Conocimientos, que permite la definición de reglas a nivel de la base de datos, y el Manejador de Objetos, que permite la definición y el manejo de los objetos.

La arquitectura de INGRES es de Cliente Multiservidor, la cual permite procesos de interfaz final en sistemas cliente para ejecutar aplicaciones, mientras que un servidor o

servidores de fondo proveen el procesamiento de base de datos. Esta arquitectura consta de tres partes principales: el servidor DBMS, los accesos al sistema (login) y el bloqueo del sistema y una facilidad general de comunicación. El servidor es de tipo Multi-Hebra, el cual permite el múltiple acceso de usuarios a la base de datos. Los servidores pueden configurarse de acuerdo a las necesidades. La fig. IV.1.1.1.2 muestra la conectividad INGRES.

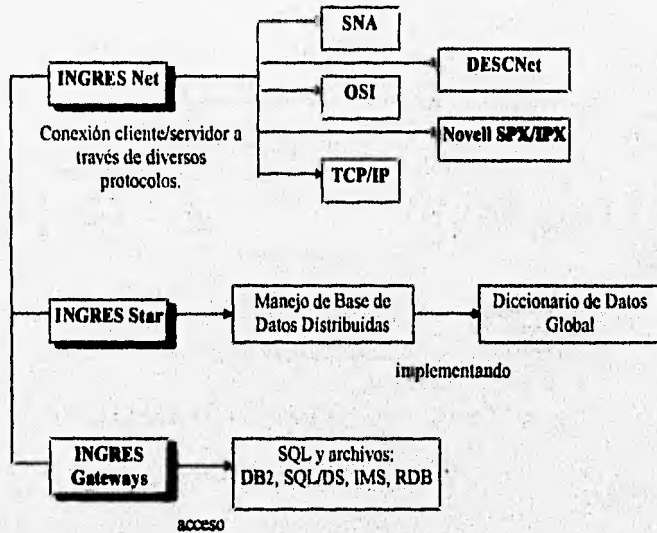


Figura IV.1.1.1.2 Conectividad INGRES

**SYBASE**

El conjunto de productos SYBASE está dividido en tres grandes áreas:

1. Servidores de Programas
2. Herramientas de ciclo de vida SQL
3. Compuertas para Programas

Existen tres versiones del servidor de programas: Servidor SQL Microsoft, Servidor SQL de seguridad y el Servidor SQL Sybase. El último de los tres es su principal producto, que corre bajo UNIX, VMS, OS/2 y NLM; Stratus es también soportado, APT es el lenguaje de cuarta generación (4GL), el cual usa el mismo diccionario de la base de datos.

Por otro lado, la herramienta CASE puede ser usada como un auxiliar para el desarrollo de aplicaciones.

La estrategia de la compuerta SYBASE es significativa, diferente a la de la mayoría de los demás vendedores, quienes sólo proporcionan una compuerta con otros productos de base de datos. SYBASE consta de dos componentes principales: el Cliente Abierto (Open Client) y el Servidor Abierto (Open Server), los cuales potencialmente proveen una ruta de migración para una serie de programas existentes y es usada para soportar el acceso a RDB, Oracle, Ingres, Informix y DB2.

El Servidor Abierto provee la capacidad para conectar SYBASE con aplicaciones cliente de sistemas que no operan con SQL por medio de un mecanismo remoto de llamadas a procedimientos, figura IV.1.1.1.3. Un producto especial de Servidor Abierto está disponible para CICS, el cual es más sofisticado que el que puede ofrecer la mayoría de los competidores, además de que hace a SYBASE más atractivo para los usuarios de mainframes IBM.

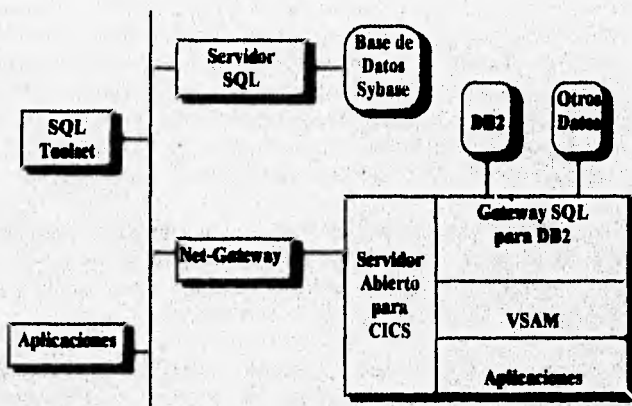


Figura IV.1.1.1.3 Conectividad SYBASE - IBM

La arquitectura de SYBASE es totalmente cliente/servidor. El manejo de datos y de las transacciones son llevados a cabo indistintamente por los usuarios y las funciones de aplicación. Esto es manejado por la ruta del "Servidor Programable", usando iniciadores, procedimientos almacenados, que facilitan el uso óptimo del ambiente de las redes distribuidas.

Las interfaces abiertas cliente/servidor (Open Client/Server) permiten a las aplicaciones correr bajo una variedad de programas de presentación y proveen acceso a otros productos bajo la sombra de SYBASE.

#### IV.1.1.2 Funcionalidad general

##### ORACLE

El grueso de la funcionalidad de ORACLE es provista por la opción Procedimental. El lenguaje PL/SQL de ORACLE es usado para escribir procedimientos almacenados, éste es un lenguaje de programación basado en Ada, con sentencias SQL fijas que añaden todas las capacidades procedimentales que le hacen falta a SQL. Este es el mismo lenguaje usado en conjunción con Oracle SQL-Forms y con los procedimientos almacenados, constituyendo de esta forma una extensión lógica del conjunto de productos.

Los procedimientos almacenados pueden ser llamados directamente por los usuarios o por los iniciadores. Los procedimientos anidados, arreglos de procesamiento, y recursión son todos soportados. Para definir de alguna forma los iniciadores, podemos decir que son procedimientos PL/SQL que ejecutan automáticamente las operaciones de INSERT, UPDATE y DELETE. Estos pueden fijarse como iniciadores antes o después de cada evento, y pueden ser disparados uno a la vez por una sentencia SQL o por un renglón. La implementación es bastante comprensiva y está basada en el estándar ANSI/ISO SQL3.

Por otra parte, ORACLE también provee alertas para la base de datos, las cuales pueden ser usadas para notificar a los usuarios de cambios críticos detectados en la base de datos. Estas pueden dispararse por un cambio predefinido de valor y pasan un mensaje a cualquier aplicación que está registrada como interesada en el evento.

El soporte para tipos de datos no estándar es limitado, aunque el almacenamiento de texto es soportado como una opción separada con una interfaz separada, pero usando la base de datos de ORACLE.

##### INGRES

En los últimos años la tendencia con las bases de datos relacionales ha sido la de fortalecer más y más su funcionalidad interna. INGRES ha sido uno de los líderes en este aspecto ofreciendo una mejor sofisticación que la mayoría de los competidores, proporcionando el almacenamiento de reglas, la definición de alertas de la base de datos y el manejo de objetos. Además de que la base de datos también puede manejar la seguridad y el control de los recursos. En la figura IV.1.1.2.1 se muestra a grandes rasgos la funcionalidad de INGRES.



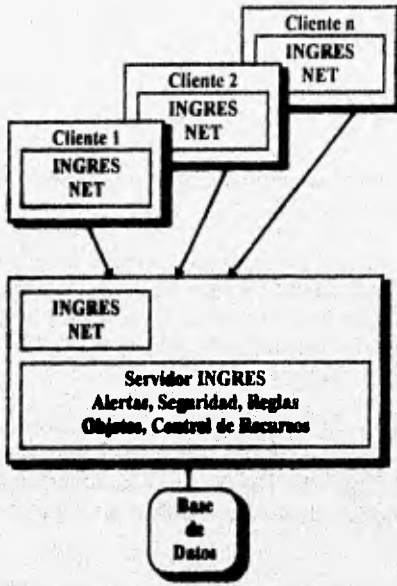


Figura IV.1.1.2.1 Funcionalidad de INGRES

El manejador de Objetos de INGRES soporta ciertos tipos de objetos orientados hacia la funcionalidad; éstos son: la definición de nuevos tipos de datos, la definición de funciones y la definición de nuevos operadores. Probablemente el más importante sea la habilidad para definir nuevos tipos de datos, refiriéndose a los tipos de datos abstractos, aunque los tres están estrechamente relacionados. El manejador de objetos permite al DBA definir nuevos tipos de datos en una sola localización.

El Manejador de Conocimientos permite al DBA definir muchas de las reglas que pueden de cualquier modo estar mantenidas en la base de datos, tal es el caso de las reglas de integridad referencial, pero también permite reglas del negocio que deberán ser definidas e impuestas. Todas las reglas y los procedimientos asociados con éstas son almacenados en la base de datos, definiéndose uno a la vez. La base de datos entonces asegura que las reglas y las acciones sean llevadas a donde sean necesarias, y el número de reglas y procedimientos que pueden ser asociados con una tabla no tiene límite.

Las alertas de la base de datos son una extensión de las reglas definidas en dicha base de datos. Cuando una regla es disparada, un proceso externo puede ser activado, esto puede resultar en un mensaje que es enviado a los usuarios interesados.

El uso de los manejadores de Objetos y Conocimientos necesita de consideraciones cuidadosas en el diseño de aplicaciones.

## **SYBASE**

SYBASE ha estado al frente del mercado en cuanto a la implementación completa de iniciadores y procedimientos almacenados.

En breve, el Servidor de la base de datos ejecutará la integridad referencial y las reglas de los negocios a través de todos los usuarios. Estas son definidas sólo una vez, almacenadas en la base de datos, y entonces usadas y compartidas por todas las aplicaciones. Esto hace la vida fácil para los desarrolladores, reduciendo el tráfico en las redes y mejorando la optimización de las consultas.

Dentro de SYBASE, el DBA está limitado a tres iniciadores por tabla (Insert, Update y Delete), pero los iniciadores pueden ramificarse en base al si-entonces-sino (if-then-else, por sus siglas en inglés) y las llamadas a procedimientos almacenados. Más aún, SYBASE coloca datos modificados dentro de una tabla y entonces permite a los iniciadores acceder la tabla.

Los procedimientos almacenados son escritos en Transact-SQL (una extensión de SQL), los cuales proveen parametrización, control lógico de flujo, llamadas anidadas y remotas, variables temporales y mensajería. El dominio de la integridad es forzado al ligar una regla para la definición del dominio usando la cláusula "Where" de SQL. La integridad de las tablas o entidades es manejada por la definición de un índice Btree único.

### **IV.1.1.3 Funcionalidad Distribuida**

#### **ORACLE**

ORACLE siempre ha provisto una Arquitectura Distribuida punto a punto (peer-to-peer) basada en una configuración de "central maestra", en la cual están basadas todas las mejoras realizadas a la versión 7, y las cuales corresponden a la opción Distribuida.

ORACLE7 está basado en la especificación XA X/Open para procesamiento de transacciones distribuidas, las que hacen compatible a este con varios tipos de monitores TP, tales como USL Tuxedo y NCR's Top-End. Esto también significa que el grabado en dos fases (two-phase commit) puede ser soportado entre ORACLE y otros productos XA compatibles, tal como es el caso de INFORMIX.

Otra parte importante de la opción distribuida es la capacidad para duplicar tablas desde una tabla maestra que se puede actualizar. Esto es, para crear y actualizar

automáticamente copias de sólo lectura (read-only). El intervalo de actualización puede ser definido por el usuario.

En general, un alto grado de transparencia de localización es provisto y las llamadas remotas a procedimientos entre nodos de la red son soportados. ORACLE también incluye otros aspectos dentro de los cuales se encuentran la administración remota de la base de datos, un nivel de optimización distribuida y soporte para nombres de la base de datos global. El juego de herramientas (toolkit, por sus siglas en inglés) está disponible para aquellos usuarios que desean proporcionar acceso SQL transparente a otras bases de datos.

## INGRES

El uso de las herramientas de INGRES, tal como el servidor Ingres Star, Ingres Gateways e Ingres/Net, permitirán a los usuarios intercambiar datos no sólo entre las bases de datos distribuidas de INGRES, sino también entre bases de datos de otros vendedores en una gran variedad de plataformas de hardware usando las aplicaciones de INGRES. Las compuertas (Gateways) ofrecen acceso transparente a otras bases de datos, tales como RDB, DB2, IMS y archivos RMS. El acceso y la manipulación de datos son manejados por una variedad de SQL que INGRES provee. Al usar OPEN/SQL en aplicaciones, el programador no tiene que preocuparse de las varias modalidades de SQL que son usadas o de los diferentes mensajes de error. Estos aspectos son manejados por la misma compuerta. Así, de esta forma al usuario siempre se le presentará una vista consistente de los datos y si es posible de los errores asociados.

El servidor Ingres Star controla los accesos tanto a las bases de datos distribuidas de INGRES como a las otras bases de datos a través de las compuertas. De esta forma, todas las bases de datos en el sistema parecerán una sola base de datos lógica para el usuario. El acceso a las tablas distribuidas es manejado por un Diccionario de Datos Central que determina dónde pueden ser localizadas las diferentes tablas a través de la red y además compila las aplicaciones. Así, el servidor Ingres Star actúa como un manejador de datos distribuido.

La capacidad proporcionada de optimización distribuida está bastante aventajada y en el ambiente distribuido la capacidad de control de recursos de la base de datos es doblemente útil. Todo esto permite al DBA asignar límites en la utilización de los recursos, en términos de CPU y de entradas/salidas a disco para los usuarios.

La parte débil de la arquitectura distribuida de INGRES es la localización centralizada del Diccionario de Datos, ya que todos los sistemas participan en el sistema distribuido necesitando acceso al Diccionario. Si el sistema responsable de la información falla, entonces la capacidad distribuida del sistema también falla.

## SYBASE

En el área general de la capacidad distribuida, SYBASE tiene aún que dirigirse a la replica, fragmentación y la necesidad de un optimizador distribuido, pero se prevé que todo esto ha sido planeado para la siguiente actualización.

La metodología por default para el grabado en dos fases, es que un servidor será designado para manejar cualquier dispositivo de almacenamiento en particular. Después verificando que todos los demás servidores estén listos para proceder, la transacción es almacenada. Pero, si uno o más de los servidores no están disponibles, entonces la transacción es automáticamente desechada. Una vez que el servidor maestro ha almacenado la transacción, esta permanecerá almacenada aún cuando uno de los otros servidores falle. El DBA no tiene elección en este procedimiento si las bases de datos están en el mismo servidor, pero fuera de eso el control programático está disponible.

La integridad de los datos está asegurada por la asociación de un identificador con cada transacción, el cual es generado por el servidor maestro. Este identificador es retenido por todos los servidores secundarios en un archivo de registro (log file). Si alguno de los servidores fallará después de que el maestro ha almacenado, entonces la transacción será recuperada en una reinicialización. Similarmente, si tanto el servidor maestro como uno de los servidores secundarios fallan, con una inconsistente en las bases de datos, entonces la situación será rectificada una vez que el servidor maestro sea reinicializado; el otro servidor entonces verificará el estado de la transacción sobresaliente y la recuperará.

Todas las actualizaciones distribuidas son manejadas por los procedimientos almacenados. El conocimiento de otras bases de datos es mantenido en la base de datos local dentro del procedimiento almacenado. Por otro lado, mientras no sean elegantes algunas soluciones (Empress Global Data Dictionary), este nunca proporcionará una localización transparente, en el sentido de que los usuarios no necesitan saber donde es realizada la actualización de los datos.

### IV.1.1.4 Desempeño

## ORACLE

La arquitectura de ORACLE proporciona buena estabilidad y con la adición de un optimizador bastante mejorado, todos los inconvenientes obvios del desempeño del producto han sido eliminados. Esto no hace que se alcance la velocidad de los productos de bases de datos para red más rápidos, pero lo hace altamente competitivo con varios de sus primos relacionales. Su escalabilidad e implementación plataformas de equipo paralelo ahora hacen a ORACLE funcional para muchos sistemas grandes.

## INGRES

INGRES es rico en estructuras de datos y opciones de puesta a tono que proporcionan un buen desempeño, especialmente para sistemas de consultas intensivas. El optimizador es uno de los más sofisticados y además proporciona una lectura adelantada, escritura diferida, depósitos agrupados y almacenamiento de múltiples archivos, la cuál es una forma eficiente de partición horizontal para tablas de gran tamaño. La única debilidad marcada en el área del desempeño, es el bloqueo a nivel de registro, el cual necesitará ajustes de diseño para los sistemas de actualización intensiva.

## SYBASE

Actualmente es la única Base de Datos Relacional diseñada específicamente para ambientes cliente/servidor maximizando el desempeño de la nueva tecnología. El desempeño de SYBASE está basado en la optimización de memoria para la búsqueda de tablas. En cuanto a su funcionamiento en ambientes grandes OLTP se mejoró su rendimiento notablemente en el SYBASE System 10. Las estructuras de datos que maneja son de alto rendimiento.

### IV.1.1.5 CPU

## ORACLE

La arquitectura de ORACLE lo hace particularmente funcional para máquinas con múltiples CPU's donde este es capaz de balancear la carga a través de los procesadores, pero las ventajas de esta arquitectura no serán tan notorias en máquinas con un solo CPU. Sin embargo los servidores son multi-hebra y el SQL compartido es un aspecto de eficiencia para el CPU. En sistemas donde las operaciones aritméticas son comunes y complejas, esto puede significar un uso extra del CPU, pero para la mayoría de las aplicaciones comerciales este defecto no será tan serio.

## INGRES

La arquitectura de multiservidores multi-hebra de INGRES permite que este sea apropiado para sistemas de múltiples CPU's. Los servidores múltiples pueden ser creados para correr concurrentemente y en paralelo tantos como CPU's estén disponibles. Alternativamente, en sistemas de acoplamiento en caso de pérdida, tal como el VAXcluster, el cual puede tener su propio servidor para acceder a la base de datos o un sistema que está diseñado como un nodo servidor, eliminan el bloqueo del tráfico de la red. En esencia, INGRES proporciona al DBA la flexibilidad para determinar dónde, cómo y en qué prioridad los servidores pueden correr.

INGRES también soporta procedimientos almacenados compilados. Estos son importantes para la actualización de sistemas intensivos donde la sobrecarga de la ejecución interpretativa no es aceptable. El código para una transacción es compilado, almacenado en la base de datos y entonces es cargado a la memoria cuando es requerido. Si es frecuentemente usado, este permanecerá en la memoria. Los procedimientos almacenados también apuntan a un plano de SQL compartido. De esta forma, el uso de la memoria es reducido debido a que el procedimiento y el plano son compartidos por todos los usuarios. La sobrecarga de una red es también reducida debido a que las llamadas son hechas directamente a procedimientos que ya han tenido varias consultas SQL a través de la red.

Por otro lado, la capacidad de INGRES para manejar tipos de datos abstractos permite que esta utilice tipos de datos nativos en todas las máquinas de la red (algunos tipos de datos tal como el "float" varían dependiendo de la plataforma de equipo). El conjunto de productos INGRES usa por sí mismo esta capacidad y contribuye a la obtención de una mayor eficiencia del CPU en una red o equipo mezclado.

## **SYBASE**

SYBASE es muy eficiente en cuanto a recursos se refiere, especialmente con sistemas capaces de atender a 50 o 60 usuarios. Los procedimientos almacenados compilados hacen el uso del CPU y la memoria más rápido y eficiente. En otras palabras, las transacciones frecuentemente usadas serán continuamente interpretadas, optimizadas y compiladas, de esta forma, el procedimiento almacenado puede ser cargado en la memoria, en donde será optimizado y entonces estará disponible para todos los usuarios. Un procedimiento almacenado sólo necesitará ser recompilado si alguno de los objetos a los cuales hace referencia cambia.

Las capacidades del multiservidor multi-hebra fueron introducidas con las versiones 4.8 y 4.9 de SYBASE, además de que este es capaz de explotar máquinas con múltiples CPU's. SYBASE se refiere a esto como una Arquitectura de Servidor Virtual, el cual utiliza la memoria compartida. Esta arquitectura muy probablemente será extendida en la próxima versión.

La tolerancia a fallas en la operación es soportada al configurar un segundo servidor en un CPU alternativo. Este servidor no usa ninguno de los recursos del CPU. Si el procesador principal en el cual está corriendo el servidor de la base de datos falla, el servidor secundario automáticamente conecta a las aplicaciones existentes, recupera la base de datos y continúa procesando.

### IV.1.1.6 Memoria

#### ORACLE

La clave del desempeño de ORACLE está en el uso efectivo de la memoria.

Otros aspectos cruciales que impactan en el desempeño de ORACLE, son la sintonización de Area Global del Sistema y el funcionamiento eficiente de los almacenes temporales. De esta forma, dependiendo de la aplicación, el funcionamiento de los almacenes temporales puede ser necesario para agrupar tablas relacionadas y así alcanzar un desempeño óptimo.

Una característica interesante es la habilidad para compartir SQL compilado entre los usuarios, disminuyendo de esta forma el trabajo de la memoria y el del CPU.

#### INGRES

Los servidores Multi-hebra permiten que la memoria sea compartida entre los usuarios. Por lo tanto, la información bloqueada, datos, procedimientos almacenados e índices son todos compartidos por los usuarios de la base de datos en el mismo sistema. El equivalente de memoria usado por el servidor es controlable ajustando el número y el tamaño de los almacenes temporales de la base de datos. Las páginas en la memoria y por supuesto su uso, son controlados por ciertos parámetros del sistema.

#### SYBASE

Los procedimientos almacenados bloquean información y datos compartidos, mientras que el servidor Multi-hebra permite que la memoria sea compartida entre todos los usuarios de la base de datos. La cantidad de memoria requerida por usuario es menor que la requerida por los productos competidores. Típicamente el Servidor SYBASE requiere 8MB, más 40K de memoria extra por usuario, lo cual es recomendable para ser altamente eficiente en la mayoría de los eventos.

La única variable para el ajuste de la base de datos disponible para el DBA es el poder variar el número de almacenes temporales a un nivel de la base de datos. En realidad es peor que esto, ya que sólo se pueden variar por servidor. Los almacenes temporales no pueden ser ajustados a un nivel de tabla y es probable que esto cause problemas, especialmente en sistemas híbridos de consulta OLTP.

#### IV.1.1.7 Entradas/Salidas a disco

##### ORACLE

ORACLE usa procesos de escritura diferida (Asíncrona) para realizar eficientemente Entradas y Salidas a disco y manejar los accesos de los usuarios al sistema (login). Permite el particionamiento horizontal para la distribución de tablas activas muy grandes sobre múltiples discos, además de que los índices pueden ser colocados en discos diferentes a los que contienen la mayoría de las bases de datos, también permite la compresión de índices mientras que la compresión de datos no.

Sin duda la mayor mejora de ORACLE7, es el optimizador de costo por estadística el cual significará una considerable reducción en las operaciones de Entrada/Salida a disco.

##### INGRES

INGRES ofrece soporte para las estructuras diferidas, llaves mezcladas, agrupación de datos y partición horizontal sofisticada para datos e índices. Si la base de datos es diseñada e implementada correctamente, los cuellos de botella son improbables al momento de consultar o actualizar sistemas con varias llamadas.

##### SYBASE

Aunque SYBASE soporta escritura diferida y almacenaje por lotes, la base de datos por sí misma está esparcida para proveer el tipo de estructuras de datos y parámetros que son típicamente requeridos para un buen desempeño en los ambientes OLTP (OnLine Transaction Processing, procesamiento de transacciones en línea). Algunas mejoras han sido implementadas en el servidor, tales como las llaves mezcladas y las listas invertidas, pero estas no están disponibles para el DBA. El tamaño de las páginas de la base de datos permanecerá invariante aunque esta sea una página de asignación optimizada para la localización en disco.

El único agrupamiento de datos soportado por SYBASE es la habilidad para almacenar datos de tablas ordenados por índice, lo cual SYBASE asegura que reducirá las fallas. Sin embargo, con sólo esta capacidad no se podrá reducir la Entrada/Salida cuando se requieran hacer uniones de tablas con relaciones "uno a muchos" en el disco, lo cual es el principal beneficio del agrupamiento



### IV.1.1.8 Ambiente de desarrollo

#### ORACLE

ORACLE proporciona un ambiente de desarrollo muy extenso, sus componentes más importantes son SQL\*Forms y SQL\*Plus. Estos comprenden las principales herramientas de desarrollo, pero otros componentes tratan con la generación de reportes, manejo de redes y facilidades de usuario final, tales como las hojas de cálculo y los gráficos.

Adicionalmente, ORACLE es comúnmente usado en conjunción con 4GLS (Lenguajes de 4a Generación) y es probablemente la base de datos más usada junto con DB2.

#### INGRES

El conjunto de herramientas de INGRES ofrece un ambiente integrado de desarrollo bastante completo, el cual consiste de Ingres/4GL, VIFRED (Visual Forms Editor), Ingres/Vision, Ingres/Forms y otros componentes. INGRES también provee Windows 4GL, que es un ambiente de desarrollo Orientado a Objetos. Todos estos productos están estrechamente asociados con la base de datos INGRES y pueden tomar ventajas del manejador de objetos y conocimiento.

#### SYBASE

El ambiente de desarrollo de SYBASE consiste de herramientas de ciclo de vida SQL, las cuales están divididas dentro de tres categorías.

El conjunto de herramientas CASE (CASE Toolset). Entre otras capacidades, posee módulos para comprender y generar esquemas para múltiples DBMS y de esta forma contribuir a la portabilidad entre SYBASE y otros productos tal como Ingres, Oracle y RDB.

El conjunto de herramientas SQL (SQL Toolset) consiste de los siguientes módulos: APT Workbench, Data Workbench, Report Workbench, Embedded SQL y SQR. Estos son usados para desarrollar aplicaciones basadas en formas, elaboración y formateo de reportes, insertar sentencias SQL en lenguajes de tercera generación (3GL) y realizar una escritura avanzada de reportes respectivamente.

Finalmente, el conjunto de herramientas de prueba SYBASE (Sybase Testing Toolset) comprende una ventaja en SQL, la cual permite editar y verificar código con facilidades de ayuda en el contexto sensitivo. El depurador SQL (SQL Debug) opera interactivamente a nivel fuente. En adición al APT, SYBASE tiene la habilidad para trabajar aproximadamente con 140 lenguajes de cuarta generación (4GLs) y generadores de reportes.

Un número de estos pueden ser herramientas para PC con interfaz al Servidor SQL. Microsoft.

#### IV.1.1.9 Utilerías

##### ORACLE

ORACLE7 provee un menú para el manejo de las herramientas del DBA (Administrador de Base de Datos, por sus siglas en inglés), constituyendo prácticamente una aplicación interfaz-final para el SQL\*DBA. Además, este permite al administrador de la base de datos añadir usuarios y/o archivos, monitorear el desempeño y el uso y realizar respaldos de la información.

El DBA puede monitorear quién está usando la base de datos, qué tablas se están accedando, cómo los bloqueos han sido generados y el nivel de la actividad de Entrada/Salida a disco. Otras utilerías soportan el respaldo de la base de datos, recuperación, exportación e importación hacia y desde las estructuras de archivos del sistema operativo.

##### INGRES

La más importante de las utilerías de INGRES es el Monitor del Desempeño Interactivo (IPM, Interactive Performance Monitor). Este es básicamente un menú para el manejo de utilerías que permite obtener estadísticas interactivas en línea las cuales serán desplegadas en la pantalla para el análisis del desempeño. El acceso es proporcionado para la utilización de los recursos, el archivo de entrada y la información del sistema de reloj.

La utilería "Starview" es provista con el servidor "Star" para permitir al DBA observar la base de datos distribuida y conocer de esta forma qué tablas están presentes sin tener acceso a la información del diccionario de datos central. Por último, también se proporcionan las utilerías para respaldos y recuperación de información (Backup y Restore), incluyendo soporte de respaldo mientras el sistema está en línea.

##### SYBASE

La cuarta parte de las herramientas de ciclo de vida SQL está constituida por el conjunto de Herramientas de Operación, el cual incluye el Sistema de Administración (SA Companion), que es usado por el DBA para instalar los servidores, manejar el espacio en disco y establecer las cuentas de los usuarios de la red. Por otro lado, el Monitor SQL provee información para la puesta a tono de la base de datos, descargar cuentas y tener la capacidad de planeación.

Data Workbench ofrece una transferencia masiva de datos y una utilidad de conversión así como también la habilidad para acceder el catálogo del sistema en donde el diccionario de datos es almacenado como una serie de tablas. Finalmente, está la utilidad que chequea las estructuras de la base de datos (encabezados de página, ligas, etc.) y asegura que todas estén funcionando correctamente.

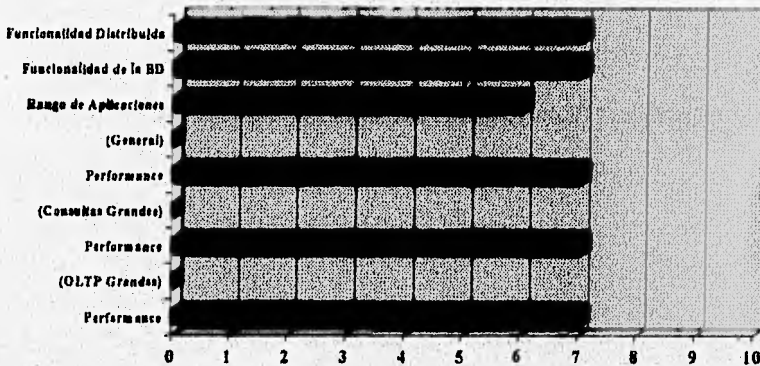
**IV.1.2 Elección de la herramienta**

Para la elección de la mejor alternativa de solución, se tomaron en cuenta los criterios mencionados anteriormente, considerando para cada uno sus ventajas y desventajas.

Los resultados obtenidos se calificaron en base a esta información y se muestran en las siguientes gráficas y tablas comparativas:

Cabe aclarar que la fuente de dónde se obtuvieron las siguientes gráficas y tablas no mencionaba las condiciones de la prueba.

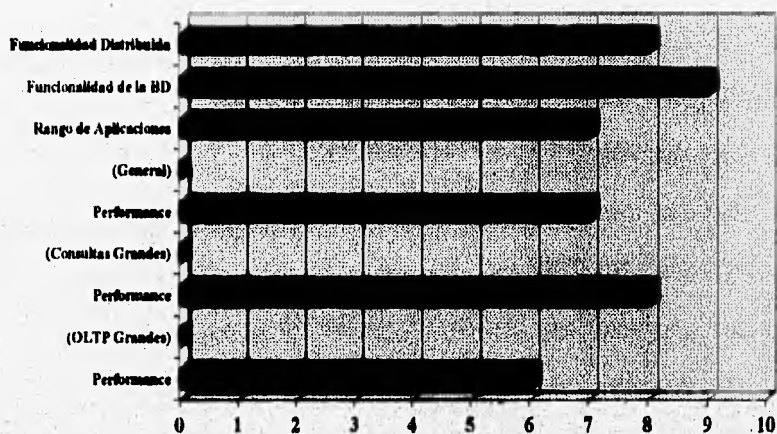
**ORACLE**



Fuente: International Data Corporation (1994)

**Gráfica IV.1.2.1 Características de rendimiento y funcionalidad de ORACLE**

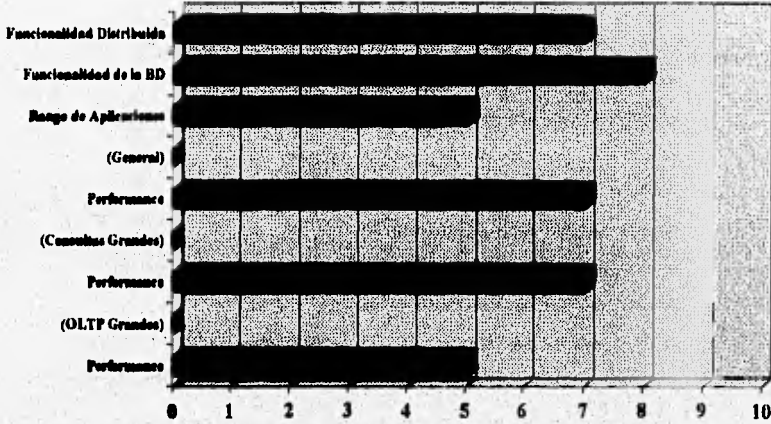
# INGRES



Fuente: International Data Corporation (1994)

Gráfica IV.1.2.2 Características de rendimiento y funcionalidad de INGRES

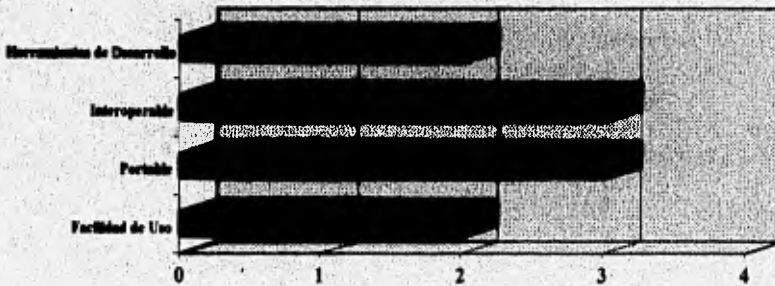
## SYBASE



Fuente: International Data Corporation (1994)

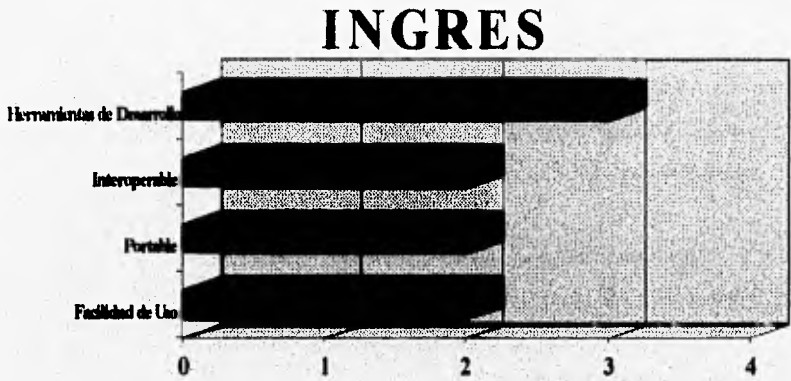
**Gráfica IV.1.2.3 Características de Rendimiento y Funcionalidad de SYBASE**

## ORACLE



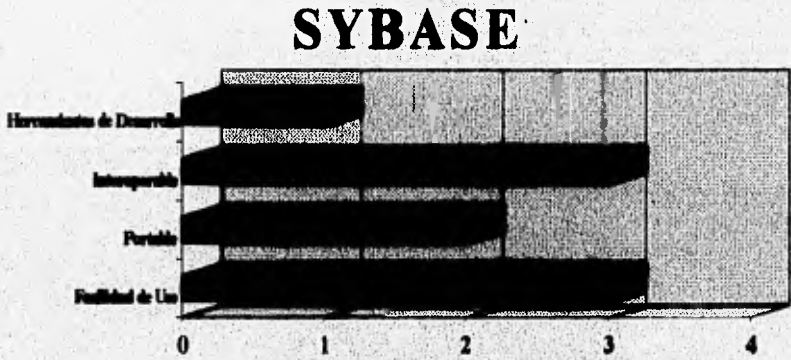
Fuente: International Data Corporation (1994)

**Gráfica IV.1.2.4 Características de manejo de ORACLE**



Fuente: International Data Corporation (1994)

Gráfica IV.1.2.5 Características de manejo de INGRES



Fuente: International Data Corporation (1994)

Gráfica IV.1.2.6 Características de manejo SYBASE

<b>FUNCIONALIDAD DISTRIBUIDA</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Acceso de lectura remota	SI	SI	SI
Acceso de Actualización remota	SI	SI	SI
Protocolo Común Bifásico	SI	SI	SI
Uniones sobre múltiples Bases de Datos (un mismo nodo)	SI	SI	SI
Uniones sobre múltiples Bases de Datos (dos o más nodos)	SI	SI	SI
Vistas sobre múltiples Bases de Datos	SI	SI	SI
Actualización de vistas sobre múltiples Bases de Datos	NO	NO	SI
Acceso transparente a la Base de Datos	SI	SI	SI
Definición de múltiples Bases de Datos físicas como una sola Base de Datos lógica en el Diccionario de datos	SÓLO EN EL DICCIONARIO DEL CASE	SI	NO
Referencia remota a Bases de Datos físicas a través de iniciadores	SI	NO	SI
Replicación automática de datos	SI	SI	NO
Transparencia en la replicación de datos	SI	SI	NO
Arquitectura distribuida igual a igual o maestro/esclavo	IGUAL A IGUAL	MAESTRO/ ES-CLAVO	IGUAL A IGUAL

Fuente: International Data Corporation (1991)

Tabla IV.1.2.1 Funcionalidad distribuida

<b>CPU Y MEMORIA</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Uso de tipos de datos nativos del equipo	NO	SI	SI
Soporte de SQL dinámico	SI	SI	SI
Compilación de procedimientos almacenados	SI	SI	SI
Configuración de múltiples procesos del servidor	SI	SI	SI
Asignación específica de usuarios a los procesos del servidor	SI	SI	SI
Soporte de multiprocesamiento simétrico	SI	SI (Via una arquitectura de multi-servidores multi-hebra)	SI (Arquitectura de servidor virtual)
Compresión de datos	NO	SI	SI
Compresión de índices	SI	SI	SI
Grupo de Llamadas	SI	SI	SI
Otros	Llamadas Rápidas (Asíncrono)	Llamadas Rápidas, Caché del servidor compartido	---

Fuente: International Data Corporation (1994)

**Tabla IV.1.2.2 CPU y Memoria**



<b>CPU Y MEMORIA</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Uso de tipos de datos nativos del equipo	NO	SI	SI
Soporte de SQL dinámico	SI	SI	SI
Compilación de procedimientos almacenados	SI	SI	SI
Configuración de múltiples procesos del servidor	SI	SI	SI
Asignación específica de usuarios a los procesos del servidor	SI	SI	SI
Soporte de multiprocesamiento simétrico	SI	SI (Via una arquitectura de multi-servidores multi-hebra)	SI (Arquitectura de servidor virtual)
Compresión de datos	NO	SI	SI
Compresión de índices	SI	SI	SI
Grupo de Llamadas	SI	SI	SI
Otros	Llamadas Rápidas (Asincrono)	Llamadas Rápidas, Caché del servidor compartido	---

Fuente: International Data Corporation (1994).

**Tabla IV.1.2.2 CPU y Memoria**

<b>ENTRADAS/SALIDAS A DISCO</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Llaves mezcladas	SI	SI	SI
Índices multinivel	SI	SI	SI
Listas invertidas	NO	NO	SI
Datos agrupados en base a las llaves	SI	SI	SI
Diseño de tablas para compartir páginas de datos	SI	NO	NO
Colocación de tablas en discos específicos	SI	SI	SI
Partición horizontal de tabla	SI	SI	SI
Partición vertical de tabla	NO	SI	SI
Variación del tamaño de la página de una tabla	NO	NO	NO
Variación de la página de un índice	NO	NO	NO
Número variable de almacenes temporales por tabla	SI	SI	NO
Número variable de almacenes temporales por base de datos	SI	SI	NO

Fuente: International Data Corporation (1994)

**Tabla IV.1.2.3 Entradas/Salidas a disco**

<b>FUNCIONALIDAD INICIADORES</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Soporte de Inicidores	SI	SI	SI
De Inserción	SI	SI	SI
De Borrado	SI	SI	SI
De Actualización	SI	SI	SI
Actualización de una columna específica	SI	SI	SI
Llamada a Iniciador	SI	NO	NO
En Cascada	SI	SI	SI
Disparo de sentencias SQL (DML)	SI	SI	SI
Disparo de sentencias SQL (DDL)	SI	NO	SI
Disparo de múltiples sentencias SQL	SI	SI	SI
Invocar procedimientos alma- cenados	SI	SI	SI
Inclusión de lógica de proce- dimiento	SI	SI	SI
Disparo de procedimientos ex- ternos	SI	SI	SI

Fuente: International Data Corporation (1994)

**Tabla IV.1.2.4 Funcionalidad en iniciadores**

<b>FUNCIONALIDAD PROCEDIMIENTOS ALMACENADOS</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Soporte de procedimientos almacenados	SI	SI	SI
Almacenados como objetos de la base de datos	SI	SI	SI
Almacenados como archivos ejecutables externos a la base de datos	NO	NO	SI
Traídos y editados por los usuarios de la base de datos	SI	SI	SI
Ejecutados via iniciador o llamada a un programa	SI	SI	SI
Ejecución directa del iniciador por el usuario	SI	SI	SI
Capacidades del lenguaje procedural	SI	SI	SI
Nivel máximo de anidamiento	SI	SI	16
Soporte para arreglos de procesamiento	SI	SI	SI
Soporte para SQL DDL	NO	NO	SI
Soporte para SQL DML	SI	SI	SI
Soporte de cursores SQL	SI	SI	NO

Fuente: International Data Corporation (1994)

**Tabla IV.1.2.5** Funcionalidad en procedimientos almacenados

<b>SEGURIDAD DE USUARIO EN LA BASE DE DATOS</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Nivel base de datos	SI	SI	SI
Nivel tabla	SI	SI	SI
Nivel de campo	SI	SI	SI
Nivel campo por valor	SI	SI	SI
Seguridad por grupo de usuario	SI	SI	SI
Nivel de vista	SI	SI	SI
Privilegios para crear procedimientos almacenados	SI	SI	SI
Privilegios para ejecutar procedimientos existentes	SI	SI	SI
Privilegios para alterar y desechar procedimientos existentes	SI	SI	SI

*Fuente: International Data Corporation (1994)*

**Tabla IV.1.2.6 Seguridad de usuario en la base de datos**

<b>RECUPERACIÓN</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Recuperación hacia adelante	SI	SI	SI
Recuperación hacia atrás	SI	SI	SI
Configuración de auto llamada para SQL	SI	SI	Default
Deshabilitar registro de transacciones	NO	SI	NO
Respaldo de la base de datos y registro de las transacciones simultáneamente	No Aplicable	SI	SI
Respaldo de la base de datos y registro de las transacciones por separado	SI	SI	SI
Respaldo en línea	No Aplicable	SI	SI
Recuperación de una porción seleccionada del respaldo	SI	SI	NO

Fuente: International Data Corporation (1994)

Tabla IV.1.2.7 Recuperación

<b>UTILERIAS</b>	<b>ORACLE</b>	<b>INGRES</b>	<b>SYBASE</b>
Monitoreo en línea de los bloques en la base de datos	SI	SI	SI
Monitoreo en línea de las Entradas/Salidas a disco	SI	SI	SI
Monitoreo en línea de verbos	SI	NO	NO
Monitoreo en línea de la memoria	SI	SI	No Aplica
Reorganización de los datos en línea	SI	SI	SI

Fuente: International Data Corporation (1994)

Tabla IV.1.2.8 Utilerías

Como se puede apreciar en las comparaciones anteriores, la opción que mejor cumplió con los criterios especificados anteriormente es el RDBMS INGRES, pero actualmente presenta un deficiente soporte técnico y comercial por la empresa que lo representa, además la información que ellos presentan no concuerda con la realidad que se publica en varias publicaciones, por lo que fue descartado y entonces se analizaron las dos opciones restantes.

De esta forma, ORACLE7 y SYBASE se analizaron y se encontró que ambos DBMS estaban muy parejos en cuanto a sus características, pero en facilidades ofrecidas, SYBASE otorga un 75% de descuento a la UNAM, debido a esto se recurrió al factor Costo-Beneficio en donde ORACLE7 resultó ser más costoso ofreciendo los mismos beneficios, motivo por el cual SYBASE fue seleccionado como la herramienta estándar para desarrollo de las aplicaciones para la automatización de los procesos administrativos de la DGENP, además de que en varias Instituciones de la UNAM se ha tomado como estándar a SYBASE.

## **IV.2 ELECCIÓN DE LA HERRAMIENTA VISUAL PARA DESARROLLO CLIENTE/SERVIDOR.**

Una base de datos cliente/servidor es una combinación de equipo y programas cuya utilidad se reduce si no se cuenta con medios de acceso a los datos. A pesar de que los proveedores de bases de datos ofrecen, muchas veces, sus propias herramientas de desarrollo, el verdadero poder de los sistemas cliente/servidor radica en la variedad de aplicaciones cliente y programas de desarrollo, -también llamados front-ends o interfaz de usuario-, disponibles por parte de terceros.

Se puede clasificar a las interfaces de usuario en cuatro grandes categorías: agregados a productos ya existentes, herramientas de desarrollo de aplicaciones, reportadores y herramientas de análisis e integración de datos.

Las herramientas de desarrollo de aplicaciones son usadas principalmente por los programadores y están diseñadas para facilitar el proceso de creación de aplicaciones personalizadas de interfaz de usuario. El presente estudio elaborado por la Subdirección de Sistemas de la Dirección de Cómputo para la Administración Académica de la DGSCA-UNAM presenta evaluaciones de productos que están integrados en esta categoría.

### **IV.2.1 Productos Evaluados**

Se utilizó como referencia para la evaluación Visual Basic 3, por ser un producto pionero en programación general en Windows. Este junto con Delphi, son herramientas simples para desarrollo ya que carecen de facilidades para la programación de proyectos a nivel corporativo, por lo que se les denominó "cliente-bajo". SQL Windows es un producto

que cuenta con herramientas de programación en grupo y capacidad de proyectos grandes, por lo que se le denominó "cliente-alto". Finalmente, se incluyeron herramientas que pueden desarrollar y ejecutar proyectos en diversas plataformas sin estar restringidas a la PC: PowerBuilder, Vision y Omnis 7, por lo que se les denominó "multiplataforma".

#### IV.2.2 Criterios de Evaluación

El estudio se centro en la facilidad para desarrollar aplicaciones cliente/servidor, en plataforma Windows, hacia un servidor de base de datos (principalmente Sybase, dado que es el manejador que se recomienda en la UNAM), ejecutándose en plataforma UNIX. Es importante la facilidad de aprendizaje y uso, poder de la herramienta y recursos que consume. En el estudio se incluyen los siguientes aspectos:

**Descripción del Producto y Características Generales:**

En esta sección se busca dar una descripción general de la herramienta, e información como nombre, marca y conclusiones de la evaluación.

**Nombre:**

Nombre Comercial

**Versión:**

Versión que se evaluó (hasta el mes de agosto de 1995 son las más actuales, aunque ya se espera la liberación de versiones superiores).

**Marca:**

Fabricante, dueño o distribuidor.

**Categoría:**

Cliente-bajo, alto-final o multiplataforma, define el alcance de la herramienta.

**Medio de Distribución:**

Dispositivo de almacenamiento en que se distribuye el producto.

**Plataformas de Desarrollo Soportadas:**

Necesidades de equipo y programas para el funcionamiento de la herramienta cuando se desarrollan proyectos. Se incluyen los requerimientos mínimos (no recomendados) y óptimos.

**Plataformas de Implantación Soportadas, Tamaño del Tiempo de Ejecución y Requerimientos en cada una:**

Necesidades para ejecutar los programas desarrollados con la herramienta.

**Ventajas:**

Pequeña descripción de las características sobresalientes del producto.



**Desventajas:**

Listado de las deficiencias de la herramienta.

**Observaciones:**

Comentarios o notas que no son directamente ventajas o desventajas.

Recomendación final.

**Utilerías Extra disponibles con el Producto:**

Herramientas adicionales que se incluyen como un producto integrado.

**Servidor de Base de Datos SQL Local:**

Manejador de Base de Datos incluido en la herramienta, con objeto de desarrollar en caso de no tener un servidor independiente.

**Reporteador:**

Programa que elabora reportes, comúnmente con librerías o "tiempo de ejecución" que permite visualizar o imprimir desde un programa hecho con la herramienta.

**Capacidades en Programación:**

Descripción del ambiente de programación.

**Interfaz con el Programador:**

¿Cómo es el ambiente? ¿Similar o diferente al resto de las aplicaciones? ¿Fácil de usar?

**CUA:**

(Common User Access o Usuarios de Acceso Común) Similitud con los demás programas de Windows.

Apego a los estándares de interfaz.

**Modo de Programación:**

Características del sistema de programación.

**Lenguaje de Programación:**

Lenguaje o dialecto que utiliza.

**Tipo:**

Grupo de lenguajes al que pertenece.

**Tipo de Código Generado:**

Tipo de código del programa final, interpretado, código-p o compilado.

**POO:**

(Programación Orientada a Objetos) Nivel de soporte que provee el producto al uso del paradigma de objetos.

**POE:**

(Programación Orientada a Eventos) Nivel de soporte al paradigma de eventos.

**Biblioteca o Repositorio de Componentes:**

Aquí se describen sus facilidades en cuanto a repositorios de componentes.

**Distribución de Aplicaciones:**

Procesos y archivos necesarios para la distribución de aplicaciones terminadas.

**Capacidades de Interconexión:**

Se describen las formas que tiene para conectarse a través de la red.

**Método de Conexión a Servidor SQL:**

Método o protocolo de comunicación que se requiere para conectarse al servidor.

**Facilidad de Migración:**

Facilidad para cambiar la plataforma de desarrollo, la de destino, y/o el servidor de base de datos.

**Desarrollo de Aplicaciones con Interconexión:**

Facilidad para comunicar los programas con otros a través del sistema operativo.

### IV.2.3 Como trabajan las Interfaces de Usuario.

Las aplicaciones cliente se ven y se ejecutan igual que cualquier otra aplicación tenga en su PC, en su Macintosh o en su estación de trabajo Unix. Si el software del cliente está diseñado de manera apropiada, el único indicio de que el usuario está usando una "Interfaz de Usuario" de un servidor remoto de bases de datos se da cuando tiene que dar tanto su clave como su contraseña o password para entrar en sesión con dicho servidor.

La secuencia de eventos que ocurren cuando el usuario accesa al servidor de bases de datos puede ser generalizada en los siguientes seis pasos:

(Cliente) El usuario crea su consulta (query).

(Cliente) La Interfaz de Usuario formatea la consulta en lenguaje SQL y la envía a través de la red hacia el DBMS.

(Servidor) El servidor de bases de datos verifica los derechos del usuario sobre los datos que desea consultar (sistema de seguridad).

(Servidor) Si se cuenta con los derechos correspondientes, el servidor de bases de datos procesa la consulta y regresa los resultados de la misma a la Interfaz de Usuario.

(Cliente) La Interfaz de Usuario recibe la respuesta y la formatea para su presentación al usuario.

(Cliente) El usuario visualiza y/o manipula los datos y/o reinicia el proceso.

La consulta o query puede ser cualquier acción que el usuario haga sobre la base de datos, como actualizaciones, inserciones, borrados, o simples consultas.

Hay que señalar que el número de herramientas cliente/servidor es inmenso, y que todas ellas tienen fortalezas y debilidades. Aun así, hay interfaces de usuario que por su poder y facilidad de uso sobresalen del resto.

Tras la evaluación, las conclusiones sobre cada herramienta son:

**Delphi ver. 1.0** : Combina la elaboración de ejecutables de alto desempeño con el primer lenguaje de dos vías, siendo una excelente opción para programadores de pascal que no cuenten con un equipo muy poderoso. La tradición Borland de herramientas de desarrollo se observa en todo su esplendor con este producto.

**Power Builder ver. 4.0** : Esta herramienta cruza un momento crucial de su historia, actualmente es una garantía de capacidad y soporte. Es muy importante para su futuro lo que Sybase logre hacer de ella, pues si logra integrarla con SQL Server y además mejorar su desempeño, sería la opción obligada para los usuarios de Sybase.

**Omnis 7 ver. 3.0** : Pese a que promete ser un ambiente de desarrollo completo, al menos lo que se logro evaluar de ella dejó mucho que desear. Quizás si se contase con una excelente plataforma de desarrollo, la capacitación necesaria y la necesidad de ejecutar las aplicaciones en Unix, Mac y Windows, pudiese considerarse como una opción, aunque posiblemente inferior a Vision.

**SQL Windows ver. 5.0** : Esta herramienta promete ser el futuro para el desarrollo de aplicaciones cliente/servidor, gracias a una facilidad de desarrollo inigualable y alianzas estratégicas. También hay que considerar los altos requerimientos de memoria requeridos.

**Vision ver. 2.0** : Una excelente herramienta multiplataforma, capaz de generar un conjunto de aplicaciones que cubran todas las necesidades de una corporación de una forma rápida y unificada a través de las diversas plataformas de equipo. Lástima que ello

impida poder explotar completamente las capacidades particulares de cada una de ellas.

**Visual Basic ver 3.0** : Esta herramienta inicio el reinado de las herramientas de desarrollo visual, y cuenta con el mayor soporte disponible en la actualidad. sin embargo, está rezagada por amplio margen con respecto a la competencia, tanto en capacidades como en desempeño, situación que pretende remediar en su próxima versión, a liberarse próximamente.

La decisión de la herramienta de desarrollo a utilizar debe estar subordinada al sistema operativo a usar, el DBMS, la plataforma de equipo con la que se cuenta y el tipo de programadores que desarrollarán las aplicaciones.

Tomando en cuenta que el sistema operativo a usar es UNIX (Solaris) se debe pensar en una herramienta con categoría de Multiplataforma, las cuales son: Power Builder 4.0 , Omnis 7 y Vision 2. Posteriormente se debe de elegir la que más compatibilice con el DBMS y con el equipo; por tal causa **se eligió Power Builder ver 4.0** , ya que además de ser de la misma compañía que el DBMS (SYBASE) posee un costo/beneficio mayor que el de su competidor más cercano, que en este caso sería Vision ver. 2.0 , pero Vision 2.0 no cuenta con Servidor de SQL Local ni con Reporteador, elementos fundamentales para el Sistema de la DGENP.

Para finalizar el capítulo, se ha incluido una tabla comparativa que presenta características interesantes de las herramientas analizadas.

Producto	Delphi 1.0	Power Builder 4.0	SQL Windows 5	Omni 7	Visica 2	Visual Basic 3
MARCA	Borland	Power Soft/ Sybase	Gupta	Blyth Software	Unify	Microsoft
CATEGORIA	Low End Client	Multiplataforma	High End Client	Multiplataforma	Multiplataforma	Low End Client
MEDIO DE DISTRIBUCIÓN	Floppies y CD-ROM	CD-ROM	CD-ROM	CD-ROM	Floppies	Floppies
PLATAFORMA DE DESARROLLO MÍNIMA	386SX, 6MB, 35 MB HD	386SX, 8MB, 19 MB HD	386SX, 8MB, 28 MB HD	386DX, 8MB (en plataforma PC)	386DX, 8MB (en plataforma PC)	386SX, 4MB, 10 MB HD
PLATAFORMA DE DESARROLLO RECOMENDADA	486, 12MB, 60 MB HD	486, 16MB, 75 MB HD	486, 16MB, 45 MB HD	486, 16 MB (en plataforma PC)	486, 16 MB (en plataforma PC)	386DX, 8MB, 40 MB HD
PLATAFORMAS DE IMPLANTACIÓN SOPORTADAS	Windows 3	Windows 3.1 Windows NT, OS/2, Mac, UNIX	Windows 3.1	Unix (Open Look y Motif), Windows y Mac	Unix (Open Look y Motif), Windows y Mac	Windows 3.1
UTILIDADES EXTRA INCLUIDAS	Herramientas para desarrollo en grupo. Constructor de consultas visuales. Código fuente de sus componentes.	PowerBuilder Enterprise Team/PDMS, Desktop y Library for Lotus Notes. Muy completa gama de herramientas.	Team Windows (control de trabajo en grupo), herramientas de control y monitoreo, compilador a C, Report Windows.	Ninguna.	Ninguna. (Es posible adquirir el manejador de Bases de Datos de Unify-Unify 2000 a un muy bajo costo).	ODBC, Controles Mejorados y Ejemplos, Visual Design Guide y la Microsoft Knowledge Base. Gran cantidad, pobre calidad.
SERVIDOR DE SQL LOCAL	Interbase SQL Solo	Watcom SQL Solo	SQL Base Solo	Ninguno	ninguno	Engine de Access 1
REPORTADOR	Report Smith	InfoMaker	Quest Reporter	Reports y Ad Hoc	Ninguno	Crystal Reports for VB
VENTAJAS	Utiliza como lenguaje de programación Object Pascal, que es simple pero poderoso. Código ampliado de mucha velocidad de ejecución. Diseñado para cliente-servidor.	PowerBuilder es la herramienta profesional más popular del mercado. Recientemente adquirida por Sybase, se puede esperar más integración con él.	Es quizá el producto más fácil de aprender, y de mayor rapidez de desarrollo, tiene una alianza importante con Microsoft. Es posible generar aplicaciones sin escribir código.	Posible migrar la plataforma de ejecución después de hacer la aplicación. Soporta una amplia variedad de formatos para gráficos.	El producto más flexible de los evaluados, es posible generar una aplicación en cualquier plataforma y ejecutarla en cualquier otra. Interface consistente.	Producto pionero y líder en programación general en Windows. Requiere pocos recursos y es rápido y fácil de aprender y usar. Cantidad ilimitada de Add-Ons.

DESVENTAJAS	Carece de facilidades para control de versiones o creación de aplicaciones de gran tamaño. Primera versión del producto.	Ambiente de programación jerárquico, un poco diferente al normal. Cambiar de dueño es una incertidumbre.	Requerimientos de hardware sumamente elevados. Su próxima versión usará controles OCX. Su interfase es poco intuitiva.	Código interpretado, de bajo desempeño. Altos requerimientos e incompatibilidad con Stackcr. No está orientado a PC's.	Su interfase choca con el CUA de la plataforma de ejecución. La interfase de uso y programación es difícil aprender.	No fue diseñado para Cliente Servidor. Muy limitado en su diseño y ha quedado obsoleto con el tiempo. Código P de bajo desempeño.
OBSERVACIONES	Una novedosa herramienta, gran velocidad de ejecución, poderoso lenguaje de programación. Borland atraviesa una etapa incierta en su futuro, pero aún así ha tenido una gran aceptación.	Quizás la mas difundida herramienta del mercado, cuenta con un gran soporte de terceros. Si Sybase sabe integrar su DBMS con ésta, la próxima versión será una excelente opción.	Se perfila para dominar el mercado de herramientas cliente-servidor, gracias a su facilidad de uso. Sin embargo, la próxima versión, ya para Windows 95, es una duda al usar controles OCX.	Presentó muchos problemas para instalarse y nunca funcionó a satisfacción. Ni para desarrollo ni para ejecución es su mejor plataforma una PC. No recomendable.	Si el problema a resolver implica el uso de varias plataformas de implantación, y el personal no está demasiado acostumbrado a la interfase, ésta, nos parece, es la solución correcta.	Punto de partida y comparación para los demás productos. Próximamente (al mes de Windows 95) se liberará la versión 4, la cual promete mejorar, pero no suficientes para ser competitivo.

Tabla IV.11.3.1 Tabla de comparativa de herramientas visuales de desarrollo cliente/servidor



# **CAPÍTULO V**

## **ESTUDIO PRELIMINAR DEL SISTEMA**





## CAPÍTULO V ESTUDIO PRELIMINAR DEL SISTEMA

### OBJETIVO DEL CAPÍTULO

Identificar los requerimientos de administración en la DGENP de la UNAM, para de esta manera poder realizar un sistema capaz de automatizar procesos que contribuyan al desarrollo de la ENP en todos sus planteles.

### INTRODUCCIÓN

La UNAM fundada en 1551, es la universidad más antigua de la América Continental, con sus cuatro y medio siglos de existencia.

Cerrada durante la guerra de independencia, La Universidad Nacional inicia su historia moderna en 1910; a partir de 1929 su nombre oficial es Universidad Nacional Autónoma de México. Por el número de estudiantes, la diversidad de carreras que imparte así como por sus actividades en investigación y culturales, es la más grande y la más universal del continente.

La administración escolar es un aspecto fundamental para el desarrollo universitario, misma que desde hace 3 décadas es apoyada por herramientas de cómputo. Se instaló la primera computadora en la sección de máquinas de la Secretaría Auxiliar, con el objetivo de atender los requerimientos de automatización del sistema de control escolar, generando las actas de exámenes y listas de alumnos, así como requerimientos de índole administrativo.

De esta manera la Universidad Nacional Autónoma de México es una de las primeras instituciones educativas en apoyar sus actividades académico-administrativas y científicas con el uso de sistemas de cómputo.

Las actividades de administración escolar se han convertido en un factor crítico ya que la Universidad Nacional Autónoma de México cuenta, en sus diferentes planteles, colegios de ciencias y humanidades, escuelas y facultades, con una población estudiantil de más de 270,000 alumnos.

Desde hace algunos años, dado el crecimiento en el número de alumnos de la UNAM, surgió la necesidad de mejorar los servicios de administración escolar, esto con el deseo de poder dar respuesta inmediata a los requerimientos de los estudiantes.

Dada la magnitud de la información y de los diferentes procesos involucrados en la administración escolar, es necesario realizar un diseño detallado para que el sistema tenga vida útil que satisfaga las necesidades presentes y futuras haciendo uso de nuevas tecnologías adecuadas y que tiendan a la descentralización de procesos e información.

---

En el bachillerato, la UNAM ofrece dos sistemas: el de la Escuela Nacional Preparatoria, fundada a fines del siglo pasado, y el del Colegio de Ciencias y Humanidades creado en 1971.

## V.1 DEFINICIÓN DEL PROBLEMA

Actualmente la Escuela Nacional Preparatoria tiene nueve planteles, ubicados en diferentes zonas del área metropolitana, a los que asisten cerca de 50,000 jóvenes estudiantes que se preparan para la vida. Como ayer, los estudios del bachillerato son fuente de polémica y de discusión académica, no podría ser de otra manera, constituyen la base formativa de los jóvenes del país presente, que han de ser aquellos que formen el país futuro.

Actualmente la Dirección General de la Escuela Nacional Preparatoria (DGENP) cuenta solamente con una red local que trabaja bajo Novell y que da servicio al edificio pero de una forma no muy eficiente, debido al número de usuarios y a la cantidad de información que se maneja, por lo que en ocasiones se vuelve muy lenta.

No se cuenta con un medio que permita la conexión o comunicación con los demás planteles, así mismo, solamente algunos de los nueve planteles cuentan también con una pequeña red local para agilizar algunas de las actividades que ahí se realizan. Con lo cual nos damos cuenta que cada plantel trabaja muy independientemente.

Para poder compartir información de un plantel a otro o de un plantel a Dirección General se hace por medio de discos y de una persona que los entregue a su destino.

En resumen, los sistemas de la DGENP se encuentran trabajando bajo una arquitectura servidor de archivos y no cuentan con una interfaz para la comunicación con las demás instituciones que son afines a ella. Automatizando estos aspectos y cambiando a otro tipo de arquitectura que se apegue más a las necesidades y a la población que se maneja, se logrará un mejor uso de los recursos tanto materiales como humanos y una reducción de tiempo considerable en los procesos y actualización de la información.

## V.2 ALCANCE DEL SISTEMA

El objetivo del estudio de este sistema es lograr la integración de la información que se maneja dentro de la DGENP, garantizando la seguridad de los procesos que se realicen.

Para lograr la comunicación entre la DGENP y cada uno de los planteles, la cual no existe hasta el momento, se propone la conexión por medio de Red UNAM. En varias instituciones de la UNAM ya se encuentra instalada la Red UNAM brindando grandes beneficios, como la confiabilidad en la transmisión de datos y la gran velocidad de acceso. En nuestro caso sería lo ideal, debido a la cantidad tan grande de información que se maneja y para que en los diversos trámites que se realizan se obtenga una respuesta inmediata (en línea).

Al contar con Red UNAM, cambiará la arquitectura de los sistemas que actualmente trabajan como servidor de archivos a cliente/servidor, para de esta manera obtener mayores ventajas en la distribución de los procesos, en la concentración de la información y en la reducción de tiempos de acceso contando con respuestas en línea.

Hoy en día en todas partes existen requerimientos y se manejan grandes cantidades de información, de ahí la necesidad de implementar sistemas que automaticen estas actividades y que ayuden a agilizar la consulta y control de la información, la DGENP no es la excepción, por eso es la preocupación de un cambio. En los últimos años la UNAM ha brindado gran apoyo a sus diferentes instituciones en cuanto al desarrollo de sistemas tanto de control escolar como administrativo teniendo en cuenta que la información que se maneja es muy delicada y no se pueden correr riesgos de que ocurran errores o pérdida de documentos y no tener ningún respaldo.

Tomando en cuenta que el principal demandante de la Universidad es el alumno; el estudio del sistema va enfocado a la automatización de procesos que son afines a él; dándole prioridad ya que es de suma importancia que exista información actualizada que pueda ser consultada en cualquier momento y llevar un control adecuado para poder emitir documentos o reportes. El que todo funcione eficazmente y sin contratiempos es vital, ya que de esto depende el crecimiento de nuestros actuales y futuros profesionistas.

El sistema deberá estar instalado en cada uno de los planteles de la Escuela Nacional Preparatoria, así como en la DGENP, para que cualquier movimiento que se realice quede registrado automáticamente y se evite la duplicación de información o alteración de la misma.

A continuación se muestran los procesos, con las situaciones críticas, explicando y proponiendo una alternativa de solución.

## V.2.1 Inscripciones

PROCESO	EXPLICACIÓN	SOLUCIÓN
Elaboración de relación de alumnos con documentos faltantes.	Registro manual de la relación de alumnos que no tienen completos sus documentos de inscripción.	Al automatizar el sistema de inscripciones a primer ingreso, se podrá generar automáticamente la relación que es necesaria para completar los documentos de todos los alumnos.
Elaboración de relación de alumnos no inscritos.	Se emite manualmente la relación de los alumnos que no concluyeron el trámite de inscripción.	El sistema, registrará a los alumnos que concluyan su trámite de inscripción, por ende podrá determinar quienes no lo hicieron.
Envío de la relación de alumnos no inscritos a DGAE.	Necesidad de dar de baja a los alumnos que no concluyeron el trámite de inscripción a primer ingreso y avisar a DGAE.	Enviar la relación a través de un archivo usando Red UNAM.
Realización de estadísticas en forma manual	Necesidad de generar diversas estadísticas de inscripciones a partir de la información disponible.	Al automatizar el sistema se podrá generar en forma automática los cálculos necesarios para la elaboración de las estadísticas.

V.2.2 Reinscripciones

PROCESO	EXPLICACIÓN	SOLUCIÓN
Dependencia del historial académico para el trámite de reinscripción.	necesidad de usar un documento oficial para que la sección de carrera pueda verificar la seriación de las asignaturas de cada alumno en su respectiva carrera.	Al automatizar el sistema de reinscripciones se encargará de verificar la seriación particular de cada una de las carreras.
Verificación de cupos en los grupos solicitados.	Necesidad de verificar el cupo en los grupos y ajustarlos en el momento de las reinscripciones, teniendo que llevar un registro manual del número de alumnos inscritos en cada uno de los grupos disponibles.	Al automatizar el sistema se eliminará el control manual de los alumnos inscritos en cada grupo.
Saturación de grupos.	Crear nuevos grupos o tener que ampliar el cupo de los mismos en el momento de las reinscripciones.	Basándose en estadísticas cuya generación sea más rápida se podrá reducir el número de grupos que se crean por saturación.
Elaboración manual de estadísticas.	Se requiere elaborar diversas estadísticas de reinscripción.	Las estadísticas podrán ser emitidas en forma automática con datos actualizados, debido al registro que el sistema hará de todas las reinscripciones realizadas.

## V.2.3 Bajas definitivas voluntarias

PROCESO	EXPLICACIÓN	SOLUCIÓN
Solicitud del alumno de realizar baja definitiva voluntaria.	Manualmente se registra la baja de cada alumno que lo solicita.	Utilizando un sistema en línea, el alumno estará dado de baja en el momento en que lo solicite.
Envío de la relación de alumnos dados de baja a DGAE.	Necesidad de enterar a DGAE de los alumnos dados de baja.	Transferir el archivo con los alumnos dados de baja, vía Red UNAM.
Elaboración manual de estadísticas de alumnos dados de baja.	Necesidad de generar estadísticas de bajas definitivas voluntarias.	Al llevar un registro de las bajas se podrán generar automáticamente las estadísticas con datos actualizados.
Entrega de un comprobante de baja definitiva voluntaria.	Se elabora de forma manual un documento para el alumno que demuestre la realización de este trámite	Basándose en la información previamente registrada el sistema imprimirá el comprobante de baja.

## V.2.4 Exámenes Extraordinarios

## V.2.4.1 Hasta dos exámenes extraordinarios

PROCESO	EXPLICACIÓN	SOLUCIÓN
Solicitud de inscripción a exámenes extraordinarios por parte del alumno.	Se revisan en forma manual las asignaturas a que tiene derecho el alumno utilizando el historial académico y las actas de calificación.	La revisión puede hacerse de manera automática por el sistema, ya que éste conocerá la seriación de cada una de las carreras, y la historia académica del alumno.
Elaboración de la relación de alumnos inscritos a exámenes extraordinarios.	Se requiere saber qué alumnos se han inscrito a cada asignatura en examen extraordinario.	El sistema estará en condiciones de emitir en forma automática esta relación porque contará con los registros necesarios.
Elaboración de estadísticas de inscripción a examen extraordinario.	Necesidad de emitir diversas estadísticas que reflejen la demanda de exámenes extraordinarios.	El sistema contará con un registro de todos los alumnos inscritos a exámenes extraordinarios que permitirá obtener dichas estadísticas.

## V.2.4.2 Mas de dos exámenes extraordinarios

PROCESO	EXPLICACIÓN	SOLUCIÓN
Solicitud de inscripción a más de dos exámenes extraordinarios por parte del alumno.	Se revisa en forma manual las asignaturas a que tiene derecho el alumno utilizando el historial académico y las actas de calificación.	La revisión puede hacerse de manera automática por el sistema, ya que éste conocerá la seriación de cada una de las carreras y la historia académica del alumno.
Autorización de más de dos exámenes extraordinarios.	La autorización es realizada por la Unidad de Administración Escolar y se entrevista a cada alumno.	El sistema consultará los datos del alumno y podrá orientar a quien tiene la facultad de autorizar.
Elaboración de estadísticas de inscripción a más de dos exámenes extraordinarios.	Necesidad de emitir diversas estadísticas relacionadas con las inscripciones a más de dos exámenes extraordinarios.	El sistema contará con un registro de todos los alumnos inscritos a más de dos exámenes extraordinarios que permitirá obtener las estadísticas.

## V.2.5 Control de Actas

PROCESO	EXPLICACIÓN	SOLUCIÓN
Registro manual de las actas que reciben de DGAE.	Es necesario registrar y llevar el control manualmente de los folios de las actas que envía DGAE para los diferentes trámites.	El sistema permitirá registrar las actas que se reciban, así como el tipo, grupo, asignatura, profesor y alumnos que incluye el acta.
Registro manual de las actas, ya llenadas, que son entregadas por los profesores a la unidad de administración escolar.	Se requiere llevar un control de las actas que ya fueron entregadas por los profesores, para que de esta manera se pueda avisar a las jefaturas de carrera de las actas faltantes y a qué profesor pertenecen.	Se llevará un registro automático, el cual facilitará el control de las actas y podremos saber en cualquier momento cuáles son las actas que faltan por entregar.
Elaboración manual de las relaciones de actas que son devueltas a DGAE.	Es necesario elaborar una relación de las actas que son enviadas nuevamente a DGAE.	El sistema elaborará automáticamente estas relaciones y serán transmitidas a DGAE a través de la red UNAM.



## V.2.6 Seguro Facultativo

PROCESO	EXPLICACIÓN	SOLUCIÓN
Verificación de que el alumno esté inscrito y de que no tenga seguro facultativo.	Necesidad de checar en el expediente del alumno que no tenga seguro facultativo y que esté inscrito.	El sistema se encargará de hacer estas verificaciones automáticamente cuando el alumno realice este trámite.
Registrar manualmente a los alumnos que solicitan el seguro y así mismo la entrega de comprobantes.	Se requiere llevar un registro manual de los alumnos que ya han tramitado el seguro facultativo y de los comprobantes que se han entregado.	El sistema eliminará este control manual de los alumnos que han tramitado el seguro facultativo.
Elaboración manual de estadísticas.	Se requiere información estadística de los alumnos que han tramitado el seguro facultativo.	Al llevar un registro automático de este trámite, la generación de estadísticas se podrá realizar en el momento necesario con la información actualizada.

## V.2.7 Registra Sanciones Académicas

PROCESO	EXPLICACIÓN	SOLUCIÓN
Se registra manualmente la sanción en el expediente del alumno	En caso de que el alumno cometa alguna falta será sancionado, para lo cual es necesario registrar la sanción en su expediente. Dicha sanción está en función a la falta que cometió.	Se registrará dicha sanción en el sistema, el cual se encargará de verificar en el momento que se realice cualquier otro trámite el tipo de la sanción y dependiendo de éste será posible o no continuar con el trámite solicitado.

**V.2.8 Emite Documentos**

<b>PROCESO</b>	<b>EXPLICACIÓN</b>	<b>SOLUCIÓN</b>
El alumno solicita la reposición del historial académico, del comprobante de inscripción, de la credencial o de alguna constancia.	Dependiendo del documento que está solicitando el alumno, se revisa su expediente y se elabora manualmente el documento.	El sistema verificará automáticamente los datos necesarios para la emisión del documento solicitado y lo imprimirá.
Emisión de etiquetas para abrir los expedientes de los alumnos.	Es necesario elaborar manualmente una etiqueta por cada alumno en la cual, se anotan los datos personales y el número de cuenta del alumno.	Al solicitarle la emisión de etiquetas al sistema, éste se encargará de imprimirlas automáticamente.

**V.2.9 Emite Estadísticas**

<b>PROCESO</b>	<b>EXPLICACIÓN</b>	<b>SOLUCIÓN</b>
Emisión de estadísticas de los diferentes procesos.	Elaborar manualmente las estadísticas de los procesos para la toma de decisiones.	Al llevar un registro automatizado de la información involucrada en los procesos, se facilitará la emisión de estadísticas, ya que se llevará a cabo, en el momento solicitado y con la información actualizada.

---

### V.3 JUSTIFICACIÓN DE LA METODOLOGÍA

Con el objeto de adaptar mejor el trabajo de investigación del sistema a las necesidades que se tienen, se elaboró un estudio de las metodologías que están orientadas a cubrir los requerimientos en cuanto a herramientas de desarrollo de sistemas. En este apartado se llevará a cabo una evaluación en cuanto a función-requerimientos, con el fin de definir aquella metodología que se apegue en su mayor parte a los requerimientos del sistema.

Las metodologías consideradas para el estudio, se listan a continuación:

- Metodología Jackson (Jackson System Development).
- Análisis orientado a los objetos.
- Prototipos.
- Análisis Estructurado.

#### V.3.1 Análisis de Metodologías

##### V.3.1.1 Metodología Jackson

El procedimiento de desarrollo de la metodología Jackson, la cual también se conoce como JSD (Jackson System Development), se divide en seis pasos. Los cuatro primeros se refieren a la creación de la especificación del sistema requerido, y los últimos dos se refieren a la implantación de dicha especificación.

###### a) Fase de Especificación.

- 1.- Paso de acción-entidad.
- 2.- Paso estructuración de entidades.
- 3.- Paso del modelo inicial.
- 4.- Paso funcional.

###### b) Fase de Implantación.

- 5.- Paso de sincronización del sistema.
- 6.- Paso de implantación.

Hay que destacar que la metodología descarta ciertas actividades como selección e implantación del proyecto, análisis costo-beneficio, procedimiento de aceptación del sistema, etc.

La metodología JSD es utilizada para desarrollar sistemas cuyos objetivos tienen una fuerte dimensión en tiempo, y tiene que realizarse un procedimiento antes de otro. A causa de este énfasis en la dimensión del tiempo, JSD es ampliamente aplicable en sistemas de control y de procesamiento de información (en línea y en lote). Todos ellos están involucrados con un mundo real en el cual la dimensión de tiempo es de central importancia.

Las ventajas que tiene la metodología JSD son:

- Genera un modelo del mundo real permitiendo que desde un principio se definan las entidades y acciones que están involucradas en el sistema, ayudando así a guiar el análisis hacia los requerimientos del usuario y poder delimitar las fronteras del sistema.
- Utiliza poca simbología lo cual ayuda a la rápida diagramación, además las estructuras de secuencia, selección e interacción son representadas en los diagramas.
- Hay mucha mayor claridad en los diagramas ya que para poder llegar a una acción sólo hay un camino o alternativa específica ya que existe una estricta secuencia en las acciones.

Las desventajas del JSD son las siguientes:

- La simbología utilizada llega a ser muy restringida cuando se está analizando un sistema de grandes dimensiones por lo que se dificulta el trabajo de diagramación al representar diferentes situaciones.
- No existe un documento que definan los datos, como un diccionario de datos en la metodología de análisis estructurado, por lo que la especificación carece de claridad al tratar de entender los diagramas.
- Carece de documentación formal acerca del sistema por lo que si surgen nuevos requerimientos o cambios en el sistema será difícil realizarlos.

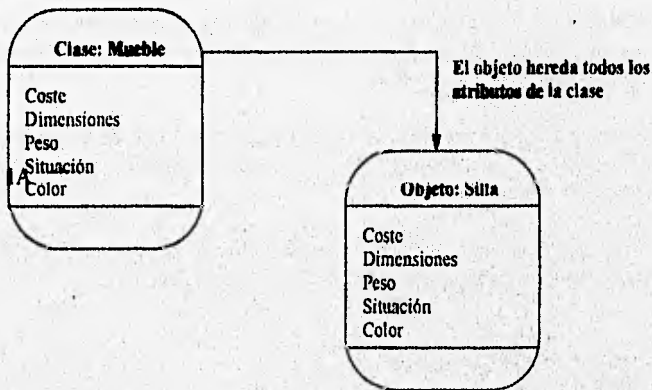
### V.3.1.2 Análisis Orientado a Objetos.

Durante mucho tiempo se han desarrollado sistemas con metodologías de análisis estructurado, sin embargo, con la idea de obtener resultados más rápidos y eficientes, hacia el inicio de la década de los 80's el concepto de "orientado a los objetos" tomaba una resonancia más fuerte y fue ganando tal prestigio, que ahora es contemplado como un enfoque para la ingeniería de sistemas.

El punto más fuerte del análisis orientado a los objetos es que es fácil de entender, ya que trabaja con conceptos conocidos desde la infancia, y que se utilizan de una manera cotidiana en la vida del hombre, entre ellos se pueden definir los siguientes:

1. Poder identificar en base a los conocimientos de las personas a los objetos y los atributos que este tiene.
2. Definir al objeto como un todo y las partes que lo componen.
3. La persona identifica perfectamente las diferencias existentes entre los diferentes objetos.

Para comprender el punto de vista orientado a los objetos, considere un ejemplo de un objeto del mundo real, una silla. Silla es un miembro (también se usa el término "instancia") de una clase de objetos mucho mayor que denominamos mueble. Se puede asociar un conjunto de atributos genéricos a cada objeto de la clase mueble. Por ejemplo, todo mueble tiene precio, dimensiones, peso, situación y color, entre muchos atributos posibles. Estos atributos, se aplican estemos hablando de una mesa o de una silla, de un sofá o de una cómoda. Dado que silla es un miembro de la clase mueble, hereda todos los atributos definidos por la clase. En la figura V.3.1.2.1 se ilustra esquemáticamente este concepto.



**Figura V.3.1.2.1 Herencia de clase a objeto**

Apoyados en el libro de Coad, Yourdon [COA90], se presenta la definición del concepto "orientado a los objetos" como sigue:

orientación a los objetos = objetos + clasificación + herencia + comunicación

dónde:

**Objeto.** Es un componente del mundo real.

**Clase.** Es un conjunto de objetos con características similares.

**Herencia.** Es la forma como los objetos adquieren todos los atributos de su clase.

Como ya se definieron algunos conceptos, ahora para poder entender de una manera más clara la metodología de orientado a los objetos, se definirá la forma en que se debe de trabajar con ella:

1. Identificación de objetos.
2. Especificación de atributos.
3. Definición de las operaciones.
4. Comunicación entre objetos.
5. Fin de la definición del objeto.

### **Identificación de Objetos**

Se puede empezar a identificar objetos<sup>1</sup> examinando la descripción del problema o llevando a cabo un análisis de la narrativa de procesamiento del sistema a construir. Los objetos pueden ser:

- *Entidades externas* (p. ej.: otros sistemas, dispositivos, gente) que producen o consumen información a ser utilizada en el sistema basado en computadora.
- *Cosas* (p. ej.: informes, visualizaciones, cartas, señales) que son parte del dominio de información del problema.
- *Ocurrencias o sucesos* (p. ej.: una transferencia de una propiedad o la terminación de una serie de movimientos de un robot) que ocurren en el contexto de operación del sistema.
- *Papeles* que juegan las personas que interactúan con el sistema (p. ej.: gestor, ingeniero, vendedor).
- *Unidades organizativas* (p. ej.: división, grupo, equipo) que son relevantes para la aplicación.
- *Lugares* (p. ej.: sala de facturación o muelle de descarga) que establecen el contexto del problema y del funcionamiento general del sistema.

<sup>1</sup> En realidad, el Análisis Orientado a Objetos lo que intenta es definir clases de las que se pueden instanciar objetos. Por tanto, cuando identificamos posibles objetos, también estamos identificando posibles clases.

- *Estructuras* (p. ej: sensores, computadoras) que definen clases de objetos o, en casos extremos, clases de objetos relacionados.

### **Especificación de atributos**

Los atributos describen un objeto que haya sido seleccionado para ser incluido en el modelo de análisis. Esencialmente, son los atributos lo que definen un objeto.

Para desarrollar un conjunto significativo de atributos para un objeto, el analista debe estudiar de nuevo la narrativa de procesamiento (o descripción del alcance) para el problema concreto y seleccionar aquello de "pertenencia" de forma razonable al objeto. Además, se debe responder a la siguiente pregunta para cada objeto. Además, se debe responder a la siguiente pregunta para cada objeto: ¿qué elementos de datos definen completamente ese objeto en el contexto del problema actual?.

### **Definición de las operaciones**

Una operación cambia un objeto de alguna forma. Más concretamente, cambia valores de uno o más atributos que están contenidos en el objeto. Por tanto, una operación debe tener "conocimiento" de la naturaleza de los atributos del objeto y deben ser implementadas de forma que se les permita manipular las estructuras de datos que hayan sido derivadas de los atributos.

Aunque existen muchos tipos distintos de operaciones, generalmente se pueden dividir en tres grandes categorías: (1) operaciones que manipulan los datos de alguna forma (p. ej.: adiciones, eliminaciones, cambios de formato, selecciones); (2) operaciones que realizan algún cálculo; (3) operaciones que monitorizan un objeto frente a la ocurrencia de algún suceso de control.

### **Comunicación entre objetos**

La definición de los objetos del contexto del modelo de análisis puede ser suficiente para establecer una base para el diseño, pero se debe añadir algo más (durante el análisis o durante el diseño) para que se pueda construir el sistema y se debe establecer un mecanismo para la comunicación entre los objetos. Este mecanismo se denomina mensaje.

La mensajería es importante para implementación de un sistema orientado a los objetos, pero no es necesario considerarla detalladamente durante el análisis de requisitos.

### **Fin de la definición del objeto**

La definición de las operaciones es el último paso para tener la especificación de un objeto completo. Se extrae la narrativa de procesamiento y mediante un análisis gramatical de las operaciones del sistema se pueden determinar operaciones adicionales, considerando el "historial" [COA90] de un objeto y los mensajes que se pasan entre objetos definidos en el sistema.

El historial genérico de un objeto puede ser definido reconociendo que se debe crear, modificar, manipular o leer en otras formas y posiblemente eliminar.

### **V.3.1.3 Prototipos**

Los sistemas muchas veces fallan en cumplir los requerimientos del usuario, esto puede darse por una falta de comunicación entre los usuarios y los analistas. Si el usuario define pobremente sus requerimientos, los resultados se trasladarán invariablemente a las especificaciones del sistema que resultarán erróneas o incompletas. Si por otra parte el analista falla al entender las verdaderas necesidades del usuario el resultado no será el adecuado.

En muchas ocasiones el uso de prototipos de trabajo ayuda a obtener y clasificar las necesidades del usuario. Esto aumenta el valor de la efectividad del ciclo tradicional del desarrollo de sistemas estructurados, especialmente durante la etapa del desarrollo. Esto permite a los usuarios tener una vista previa de su sistema real, proporcionándoles la oportunidad de experimentar con este antes de que se invierta demasiado esfuerzo en construir el sistema real. Con un prototipo, el usuario puede ver en realidad lo que llegará a ser posible; así mismo ve como se traducen sus necesidades en equipo y sistemas.

El prototipo es un sistema de trabajo diseñado para que se pueda modificar con facilidad, la información que se obtiene a través de su uso implica un diseño, que puede utilizarse a su vez, como el prototipo para obtener más información. El proceso se repite tantas veces como sea necesario para revelar los requerimientos del usuario.

En pocas palabras no es nada más un modelo vivo de un sistema que enfatiza las interfaces del usuario. Este modelo es construido para experimentación, para obtener información válida de las necesidades del usuario y para obtener una confirmación positiva de los requerimientos iniciales del sistema.

### **Fases de la metodología o ciclo de vida del prototipo**

El ciclo del prototipo es iniciado con la identificación de las necesidades del usuario que requieren una solución automatizada. El prototipo con la ayuda de los usuarios deriva



rápida de esas necesidades a los requerimientos preliminares del sistema, que serán usados para construir un modelo del sistema propuesto.

Una vez que estas actividades preparatorias son terminadas, una primera revisión del modelo de trabajo es desarrollada, con el objetivo de satisfacer tanto como sea posible la identificación funcional de los requerimientos. El prototipo es entonces demostrado en su contexto a la comunidad usuaria.

Durante la demostración, la retroalimentación con el usuario es importante para refinamientos y requerimientos adicionales. Basado en las entradas definidas por el usuario, el prototipo es rápidamente puesto al día para reflejar los requerimientos. Los ciclos de revisión son repetidos tantas veces como sea necesario hasta que el usuario este completamente satisfecho con el modelo.

### Aplicaciones de los prototipos

Algunos analistas de sistemas limitan el uso de prototipos al desarrollo de sistemas que son pobremente definidos, otros opinan que el uso de prototipos tiende a mantener proyectos pequeños y más manejables. Finalmente se cree que los prototipos se aplican mejor en las interacciones simuladas entre el usuario y el sistema.

Existen casos donde el uso de prototipos no es recomendable, por ejemplo en proyectos que se realizan en corto tiempo y el desarrollo del prototipo toma la mitad de ese tiempo.

El uso de prototipos es recomendable en el desarrollo de sistemas en línea que son difíciles de visualizar por el analista de sistemas, es una manera sencilla de visualizar el flujo de información entre pantallas.

Algunas ventajas en el uso de prototipos son las siguientes:

- La elaboración de prototipos estimula y requiere de una activa participación del usuario final.
- Se ha dicho que los usuarios finales no conocen completamente sus requerimientos hasta que los ven implementados.
- Los prototipos son modelos activos, con los que los usuarios finales pueden experimentar.
- Un prototipo aprobado es equivalente a un trabajo de una especificación de diseño en papel, con una excepción, los errores se pueden detectar mucho más temprano.
- La elaboración de prototipos puede incrementar la creatividad a través de una rápida retroalimentación con el usuario final que puede dirigir a mejores soluciones.

Algunas de las desventajas se presentan a continuación:

- Los usuarios como los analistas pueden considerar el prototipo como un sistema concluido, cuando de hecho no lo es.
- La elaboración de prototipos no elimina las necesidades para la supervisión y estudios de fases del ciclo de vida de un sistema.
- No se puede sustituir completamente cualquier prototipo por una especificación de papel. Ningún ingeniero podría realizar un prototipo sin algo de diseño de papel. La elaboración de prototipos debería ser usada para complementar, no remplazar, otras metodologías. El nivel de detalle requerido en el diseño de papel puede ser reducido, pero lo cierto es que no puede ser eliminado por completo.
- Cuando elaboramos prototipos, el alcance y complejidad del sistema se puede expandir rápidamente más allá de los planes originales. Esto puede fácilmente quedar fuera de control.

#### V.3.1.4 Análisis Estructurado

La metodología de Análisis Estructurado, tiene sus orígenes en los trabajos realizados por Tom de Marco en 1978, Gane y Sarson en 1977 y Weinberg en 1978, estos autores comenzaron sus trabajos en lo que ahora se conoce como el Análisis Estructurado Clásico, que aún cuando a principios de los 80's se vieron muchas deficiencias en dichos trabajos, fueron de gran utilidad en su momento y sirvieron de base para dar cabida a lo que ahora se conoce como el Análisis Estructurado Moderno propuesto por Edward Yourdon.

La metodología de Análisis Estructurado Moderno o Metodología YOURDON es una metodología de desarrollo de sistemas que incluye:

- Definición y estructura de los modelos estándares de sistemas y componentes del sistema.
- Notaciones, estándares y reglas para las herramientas de construcción de modelos.
- Técnicas, heurísticas y estrategias para la aplicación de la metodología Yourdon a los problemas del mundo real.

El objetivo de la metodología Yourdon es representar un sistema en un modelo esencial.

El modelo esencial del sistema es un modelo de lo que el sistema debe hacer para satisfacer los requerimientos del usuario. Durante el análisis no se describe el medio físico en el cual se almacenarán los datos (disco, cinta, etc.) y la tecnología en la cual se desarrollará e implantará. El modelo esencial está compuesto de:

- Un modelo ambiental.
- Un modelo de comportamiento.
- Un modelo de información.

**Ventajas:**

- El análisis estructurado se basa fundamentalmente en el desarrollo de modelos abstractos para representar la realidad y a través de éstos se identifican las necesidades del usuario.
- La metodología es gráfica lo cual permite el fácil entendimiento tanto de las personas involucradas en el desarrollo del sistema como de los usuarios, lo cual asegura la construcción del sistema correcto.
- Es particionable, es decir, permite estudiar al sistema por partes ayudando a administrar el proyecto tanto en presupuesto como en escalas de tiempo y además reduce los costos de mantenimiento.
- El uso de esta metodología como de sus herramientas ayuda a evitar la redundancia innecesaria en el desarrollo de sistema, por que facilita el discutir cambios y correcciones a los requerimientos a bajo costo y con un riesgo mínimo.
- Incluye herramientas que permiten el estudio del sistema desde el punto de vista del manejo de datos, el funcionamiento interno del sistema y el comportamiento externo con el medio ambiente.

**Desventajas:**

En el desarrollo de sistemas transcurre mucho tiempo entre el planteamiento de los requerimientos y la entrega del sistema completo, por lo que muchas veces no se llega a satisfacer las necesidades del usuario ya que los sistemas evolucionan a través del tiempo.

El análisis estructurado no permite la creación de un modelo previo al sistema real, que de oportunidad al usuario de experimentar con él antes de que se invierta demasiado tiempo y dinero en la construcción de un sistema y de este modo detectar fallas y permitir modificaciones en la etapa de diseño.

**V.3.2 Metodología Seleccionada.**

Con base al análisis de a las distintas metodologías se encontró que la metodología Jackson no cuenta con todos los elementos para hacer un estudio amplio que permita la especificación completa de los requerimientos del usuario debido a que es una metodología que tuvo su auge en la década de los setenta y no ha tenido actualizaciones posteriores. Una desventaja considerable es que el usuario no interactúa directamente en el desarrollo del sistema, por tales motivos se considera que no es adecuado a los requerimientos planteados del sistema.

En cuanto a la metodología de análisis orientado a objetos encontramos que se adecua a los requerimientos de nuestro sistema, sin embargo por ser una metodología un tanto nueva en nuestro medio no ha tenido un uso generalizado que permita un amplio conocimiento en las instituciones en donde se podría utilizar el proyecto.

El uso de prototipos ha llegado a ser muy utilizado pues permite una alta interacción con el usuario, lo cual lleva a entender perfectamente sus requerimientos y elaborar un adecuado modelo del funcionamiento del sistema en poco tiempo. Pero no existe una documentación formal que permita dar un mantenimiento para poder adecuar nuevas necesidades del usuario provocando altos costos en dichas modificaciones.

En conclusión encontramos a la metodología de análisis estructurado como la más adecuada a las características de estudio por que existe una generalidad de su uso en varias instituciones, lo cual asegura que pueda ser de fácil mantenimiento en el momento de llevar a cabo alguna modificación sobre el sistema que ya existe y una documentación formal del estudio realizado. En esa documentación se encuentran representaciones esquemáticas que ayudan a una mayor comprensión del sistema tanto para los analistas como para el usuario u otras personas que se vean involucradas en el proyecto. Así mismo resulta muy útil el enfoque que tiene la metodología para tratar al sistema desde tres puntos de vista: el primero, un punto de vista interno, es decir, la forma en como se llevan a cabo las funciones; el segundo, un punto de vista externo, es decir, como responde el sistema a su medio ambiente; y el tercero, el punto de vista de la información, que es donde se ve la forma en que son manejados los datos.

Otro punto a recalcar de esta metodología es que existe interacción con el usuario pero no la suficiente como para entender claramente sus requerimientos, por lo cual consideramos que un adecuado complemento a esta metodología es el uso de prototipos, lo cual ayuda a que el usuario intervenga directamente en el desarrollo del proyecto llegando a mejores soluciones.



# **CAPÍTULO VI**

## ***ANÁLISIS Y PROPUESTA DEL SISTEMA***



## CAPÍTULO VI ANÁLISIS Y PROPUESTA DEL SISTEMA

Para obtener buenos resultados, y sobre todo para satisfacer plenamente las necesidades que presenta el usuario, es necesario realizar un excelente análisis. En esta etapa se recopila toda la información que se relacione con el trabajo del sistema; documentos de entrada-salida; entrevistas y encuestas con la gente que de alguna forma tiene que ver con los procesos que se llevan a cabo para detectar las partes de mayor problema y retomar ideas para la innovación de una nueva herramienta.

### OBJETIVO DEL CAPÍTULO

En este capítulo, basándose en una metodología de análisis, se elaborarán documentos y diagramas que ayudarán al mejor entendimiento del desarrollo del sistema y que a su vez servirán para implementaciones futuras.

Hoy en día se cuenta con diversas metodologías para llevar a cabo un análisis de este tipo, en particular, se decidió hacer uso del análisis estructurado.

De acuerdo a la metodología, el análisis se divide en tres partes principales:

- **Modelo Ambiental.** Se compone de una definición breve pero clara del objetivo del sistema, un diagrama de contexto en donde se representa la relación que tiene el sistema con entidades del exterior y de una lista de eventos, en donde cada evento es un estímulo que hace que el sistema responda.
- **Modelo de Comportamiento.** Lo integran un conjunto de Diagramas de Datos que modelan cada proceso del sistema, describiendo el flujo y almacenamiento de los datos, aquí cabe señalar que para los fines del presente trabajo, únicamente se descendió a un nivel de detalle en los módulos principales del sistema. El modelo de Comportamiento también se compone por miniespecificaciones en las que, a nivel de pseudocódigo, se describe el trabajo que cada proceso primitivo debe realizar, finalmente se presenta el Diccionario de Datos en el que se define la composición de cada elemento utilizado en el análisis.
- **Modelo de Información.** En este modelo se presenta el Diagrama de Entidad-Relación, mismo que ilustra cuáles son y como se relacionan las entidades que forman parte del sistema. Por último, se presenta una descripción detallada de los atributos de cada entidad.



## VI.1 MODELO AMBIENTAL

“El primer modelo importante que se debe desarrollar como analista es uno que sólo defina las interfaces entre el sistema y el resto del universo, es decir, el ambiente.” [YOU93]

Además de determinar que está en el interior del sistema y que en el exterior, también es importante definir que información produce como salida al ambiente externo. Ningún sistema de información toma todos los datos disponibles en el universo, ni produce salidas al azar.

“Los sistemas tienen como propósito producir salidas como respuesta a algún acontecimiento o estímulo en el ambiente. Así, otro aspecto crítico del modelo ambiental consiste en identificar los acontecimientos que ocurren en el ambiente al cual debe responder el sistema. Sólo nos preocupan aquellos que ocurren en el ambiente exterior y requieren una respuesta del sistema.” [YOU93]

Este modelo define el alcance del sistema. Considerando que un sistema es un conjunto de respuestas planteadas a eventos que ocurren en el ambiente, necesitamos definirlo, es decir, la razón por la que el sistema existe (objetivo del sistema), lo que está afuera del sistema y con lo que hay una interacción (diagrama de contexto), y lo que sucede y provoca que el sistema responda (lista de eventos). El modelo ambiental está formado por tres componentes:

### VI.1.1 Objetivo del Sistema

“Es una declaración textual, breve y concisa del propósito del sistema, dirigida al nivel administrativo superior, la administración de los usuarios, y otros que no están directamente involucrados con el desarrollo del sistema.” [YOU93]

El objetivo del sistema describe por qué el sistema existe y por lo tanto, proporciona una guía para definir tanto el ambiente como las respuestas del sistema. Puede constar de una, dos o varias frases pero no debe llegar a más de un párrafo, ya que la intención no es proporcionar una descripción completa y detallada del sistema.

El objetivo del sistema es lograr la integración de la información que se maneja dentro de la DGENP, garantizando la seguridad de los procesos que se realicen; así como lograr la conexión de Dirección General con sus planteles y demás entidades externas.

### VI.1.2 Diagrama de Contexto

El diagrama de contexto es un caso especial del diagrama de flujo de datos, en donde un solo círculo representa todo el sistema y los rectángulos son las entidades externas con las que el sistema tiene alguna relación.

El diagrama de contexto enfatiza varias características importantes del sistema:

- Las personas, organizaciones y sistemas en el ambiente con los que se comunica el sistema, se conocen como terminadores y se representan con un rectángulo.
- Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
- Los datos que el sistema produce y que se envían al mundo exterior.
- Los archivos de datos que el sistema comparte con los terminadores.
- La frontera entre el sistema y el resto del mundo.

El diagrama de contexto no describe los detalles internos del sistema.

#### **Notación del diagrama de contexto**

**Entidad:** es la gente y/u otros sistemas con los que hay comunicación.

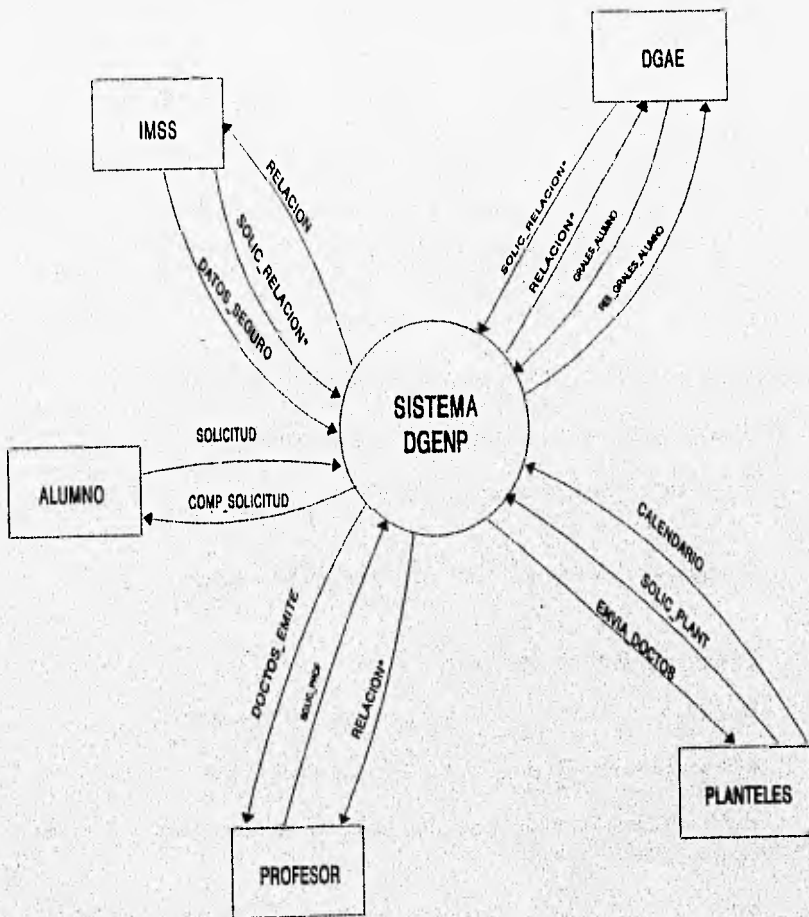
**Proceso sencillo:** nombrado para describir lo que el sistema hace.

**Flujos:** muestran las comunicaciones de los datos entre una entidad y el sistema, provocando una respuesta inmediata.

**Archivos:** muestran datos almacenados por una entidad y que son necesitados después por el sistema o viceversa.

En la siguiente figura se muestra el diagrama de contexto del sistema para la DGENP.

### DIAGRAMA DE CONTEXTO



### VI.1.3 Lista de Eventos

“La lista de eventos es una lista narrativa de los estímulos que ocurren en el mundo exterior a los cuales el sistema debe responder.” [YOU93]

La lista de eventos describe los acontecimientos o sucesos que ocurren en el ambiente y que ocasionan que el sistema responda.

Debe contener las cosas que ocurren en el ambiente y que provocan que el sistema responda y quién o qué inició el evento.

#### Lista de Eventos

1. DGAE envía relación de alumnos aceptados a inscripción.
2. Alumno solicita inscripción.
3. DGAE envía documentos de inscripción.
4. DGAE solicita relación de alumnos activos.
5. Profesor solicita relación de alumnos inscritos por grupo.
6. Alumno solicita examen extraordinario.
7. DGAE solicita relación de alumnos inscritos a examen extraordinario.
8. Planteles envían calendario de exámenes extraordinarios.
9. Planteles solicitan relación de pago a sinodales.
10. Profesor solicita relación de alumnos inscritos a examen extraordinario.
11. Alumno solicita rectificación de calificación.
12. DGAE envía actas.
13. Profesor solicita actas.
14. Profesor entrega actas calificadas.
15. DGAE solicita actas calificadas.

- 
16. Alumno solicita baja definitiva.
  17. Alumno solicita constancia de estudio.
  18. Alumno solicita reposición de historia académica.
  19. Alumno solicita reposición de comprobante de inscripción.
  20. Alumno solicita actualización de sus datos personales.
  21. Plantel (n) solicita sanción académica al alumno.
  22. Alumno solicita seguro facultativo.
  23. IMSS solicita listado de alumnos que solicitaron seguro facultativo.
  24. IMSS proporciona los datos del seguro facultativo.
  25. DGAE solicita relación de alumnos sobresalientes.
  26. La jefatura de administración escolar solicita al departamento correspondiente las estadísticas de inscripciones y reinscripciones.
  27. La jefatura de administración escolar solicita al departamento de estadísticas de exámenes extraordinarios.
  28. La jefatura de administración escolar solicita al departamento de estadísticas las estadísticas de otros ingresos a la DGENP-UNAM
  29. La jefatura de administración escolar solicita al departamento de estadísticas las estadísticas varias.

## VI. 2 MODELO DE COMPORTAMIENTO

Posterior al modelo ambiental debemos construir el modelo de comportamiento. Esto involucrará el desarrollo de diagramas de flujo de datos, miniespecificaciones para los procesos primitivos y el diccionario de datos.

“El modelo de comportamiento describe el comportamiento que del sistema se requiere para que interactúe de manera exitosa con el ambiente.”[YOU93]

Esto implica dibujar el borrador del diagrama de flujo de datos, con procesos para la respuesta del sistema, balanceados ante cada acontecimiento que se identificó en la lista de eventos. A continuación se dibujarán los archivos y finalmente se conectarán los flujos de entrada y salida apropiados a los procesos y se comparará el conjunto de diagramas de flujo de datos con el diagrama de contexto para asegurar la consistencia.

Al tener todos los diagramas de flujo de datos nivelados procederemos a realizar las miniespecificaciones únicamente de los procesos primitivos, es decir, de aquellos en los que ya no hayamos descendido otro nivel. En paralelo debemos elaborar el diccionario de datos.

### VI.2.1 Diagramas de Flujo de Datos

El trabajo básico realizado por los sistemas es la transformación de entradas en salidas.

“El diagrama de flujo de datos es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.”[PRE93]

El diagrama de flujo de datos es una herramienta que exhibe un sistema como una red de procesos conectados uno al otro por flujo de datos. El propósito es poner atención en los datos y en las transformaciones. El control, la secuencia o los algoritmos procedurales no deben aparecer en estos diagramas.

Los diagramas de flujo de datos están formados por procesos, flujos de datos y archivos.

Un proceso cambia flujos de entrada en flujos de salida, está representado por un círculo y su nombre debe describir lo que el proceso hace.

El flujo representa un paquete de datos en movimiento, es representado por una flecha que muestra la dirección del movimiento del paquete. Un paquete puede ser de datos o materiales. Cada flujo puede contener sólo un paquete, el nombre del flujo debe describir el significado del paquete.

El archivo es una colección de paquetes (registros) que están en reposo. Se representa por un par de líneas paralelas que delimitan el nombre del archivo.

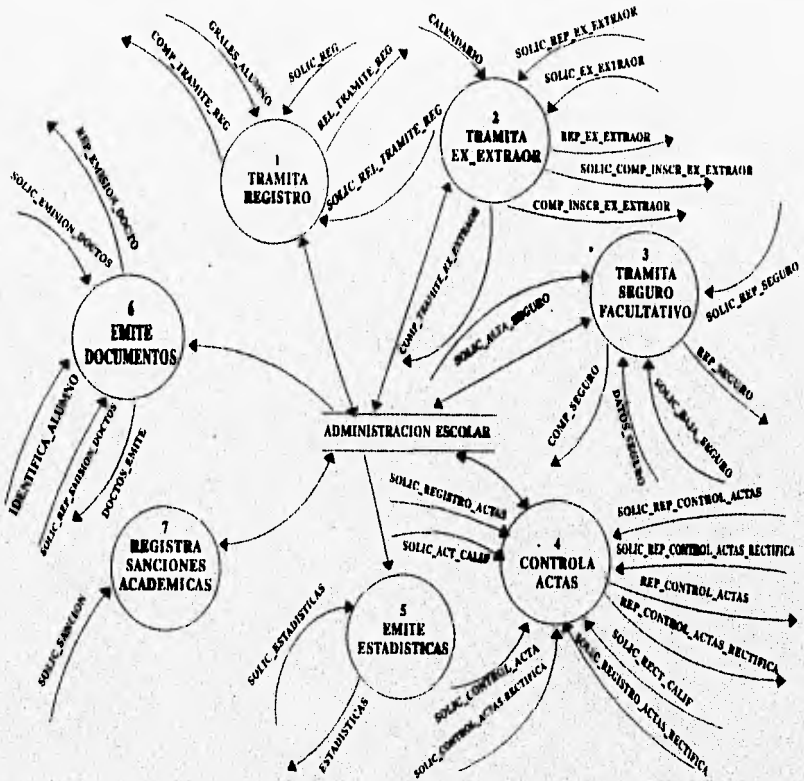
Existen ciertas convenciones para los diagramas de flujo de datos, las cuales deben facilitar las tareas de creación y revisión de los mismos.

- Cada proceso debe tener al menos un flujo de datos que entra a él. Puede ser un flujo que venga de otro proceso o un flujo proveniente de un archivo.
- Cada proceso debe producir al menos un flujo de datos para otro proceso, para un archivo o como salidas del sistema. Asimismo, deben producir nuevos datos o hacer modificaciones.
- Cuando los datos son creados por un proceso y utilizados por otro, pero no en forma inmediata, deben viajar a un archivo, de donde serán tomados por el proceso que los necesita.
- No hay un límite técnico para el número de flujos que van o que salen de un proceso o de un archivo. Para evitar la dificultad de usar el diagrama, se pueden agrupar flujos en un nivel superior, convirtiendo diferentes flujos en uno sólo. No importa que tan complicado sea el sistema real, el modelo que se maneja puede ser rediseñado para facilitar las cosas.
- Cuando un flujo llega a un archivo, esto significa que el archivo está siendo actualizado. Lo anterior implica que el archivo no es el mismo después que el proceso ha terminado.
- Cuando un flujo sale de un archivo podemos decir que se trata de una lectura.
- En el caso de una actualización a algún registro de un archivo, ésta se representa por una flecha hacia el archivo. Una flecha con doble punta significa que los datos que se han leído son modificados en el proceso y que se vuelven a escribir en el archivo.
- Se utilizan flujos no etiquetados, cuando una instancia completa del paquete fluye hacia o desde el almacén.
- El asterisco, "\*", se utiliza para mostrar iteración.

Aunque el diagrama de flujo de datos proporciona una visión global bastante conveniente de los componentes funcionales del sistema, no da detalles de éstos. Para mostrarlos se ocupan dos herramientas textuales de modelado adicionales: las miniespecificaciones y el diccionario de datos.

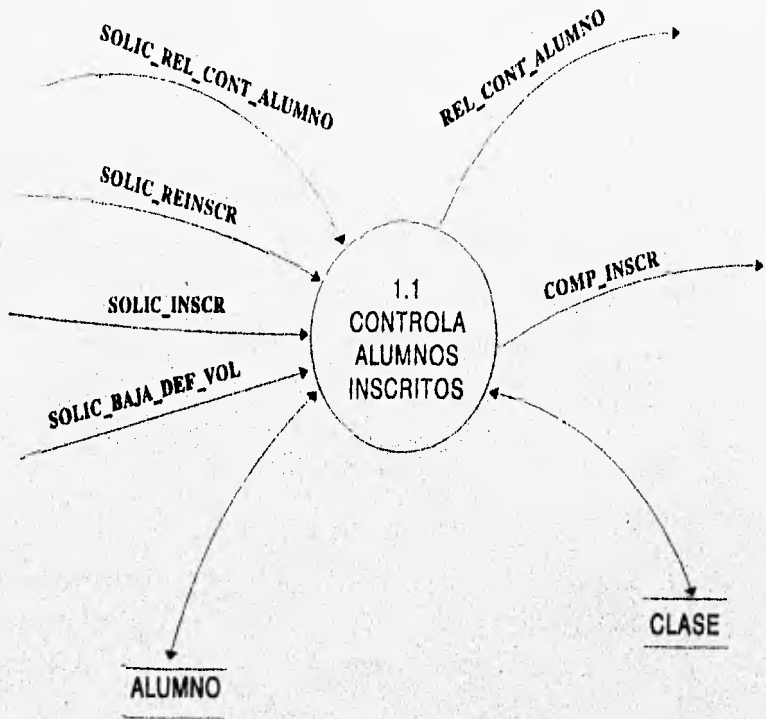
A continuación se presentan los diagramas de flujo de datos que se desprenden del Sistema para la DGENP.

**D.F.D. 0**  
**SISTEMA DGENP**

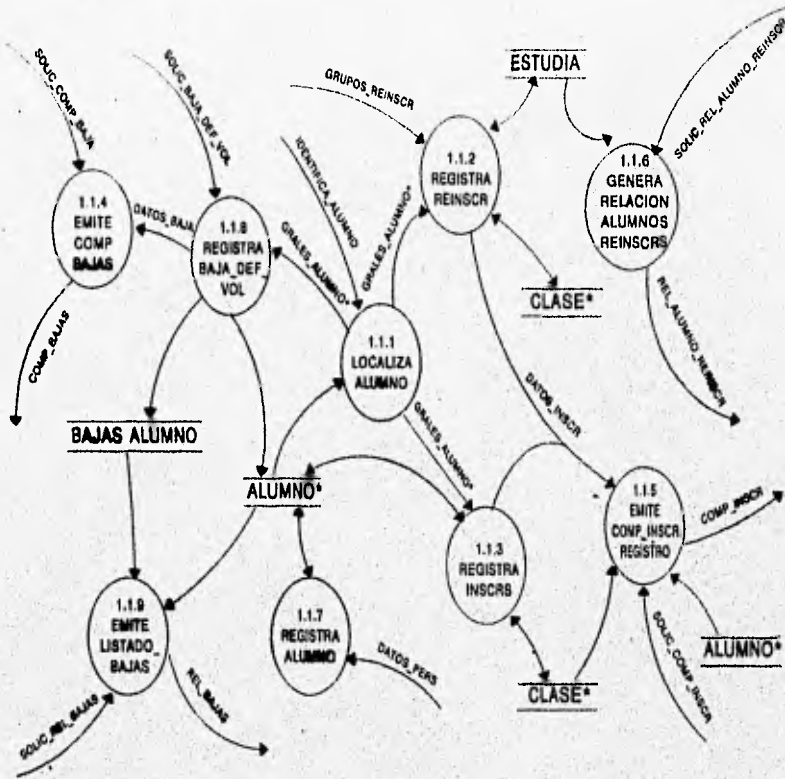




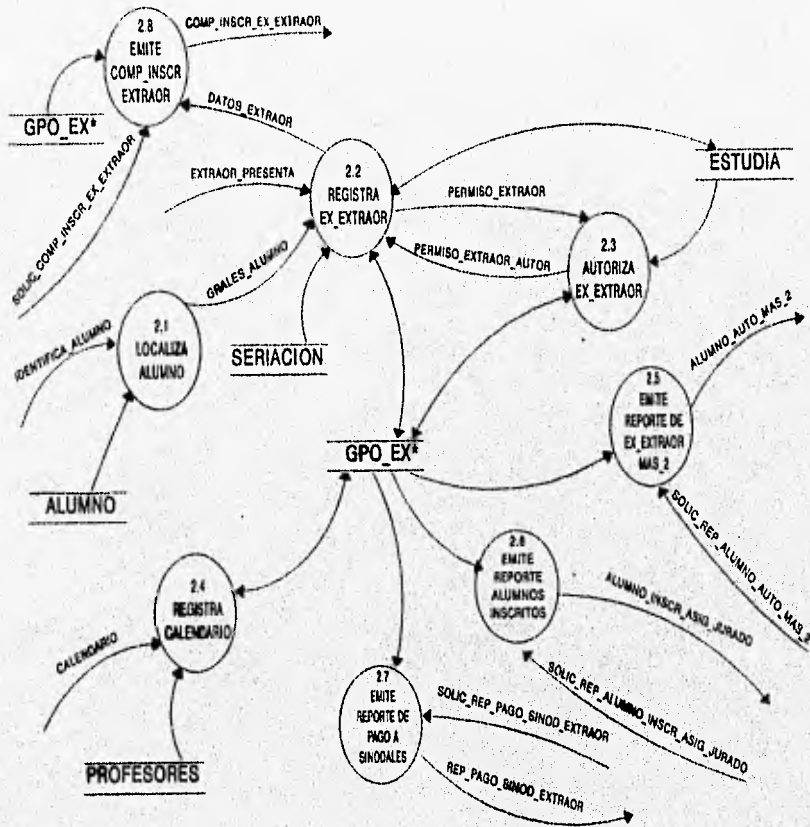
D.F.D. 1  
TRAMITA REGISTRO



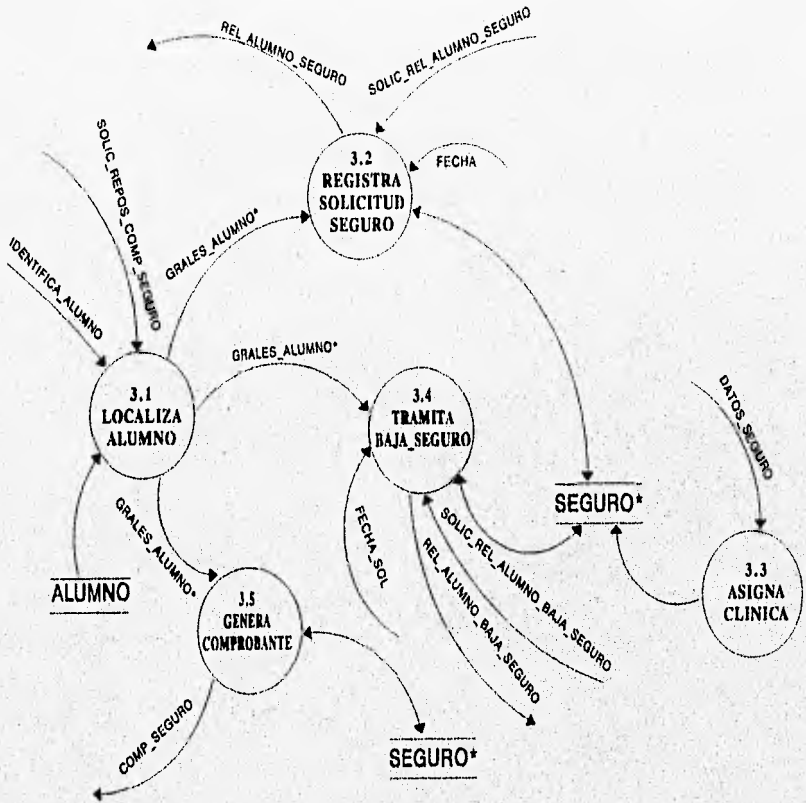
### D.F.D. 1.1 CONTROLA ALUMNOS INSCRITOS



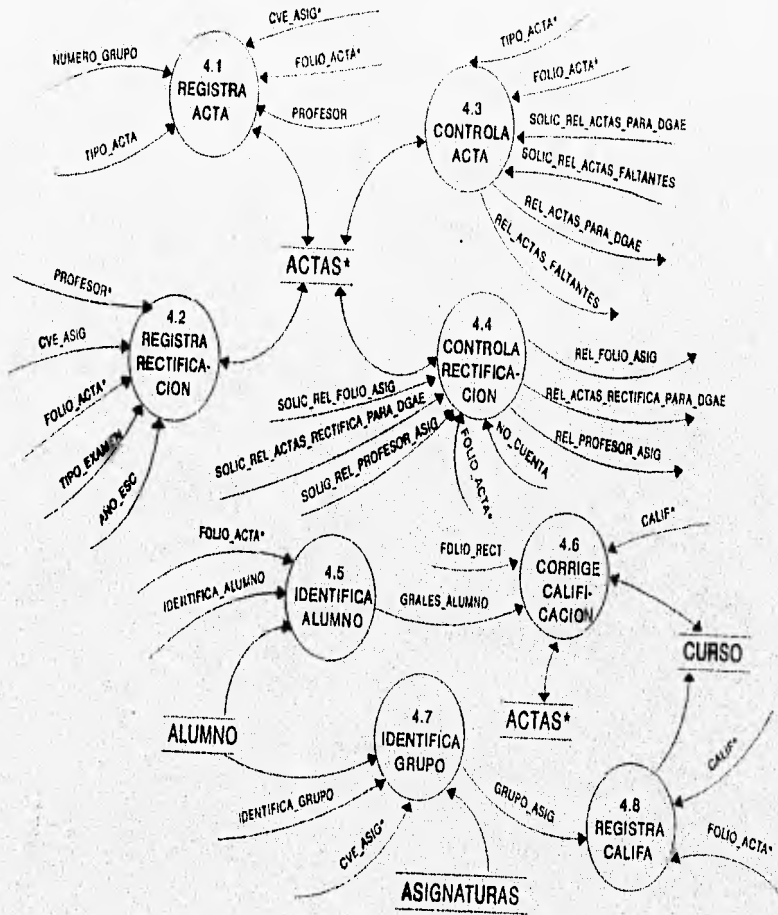
### D.F.D. 2 TRAMITA EX\_EXTRAOR



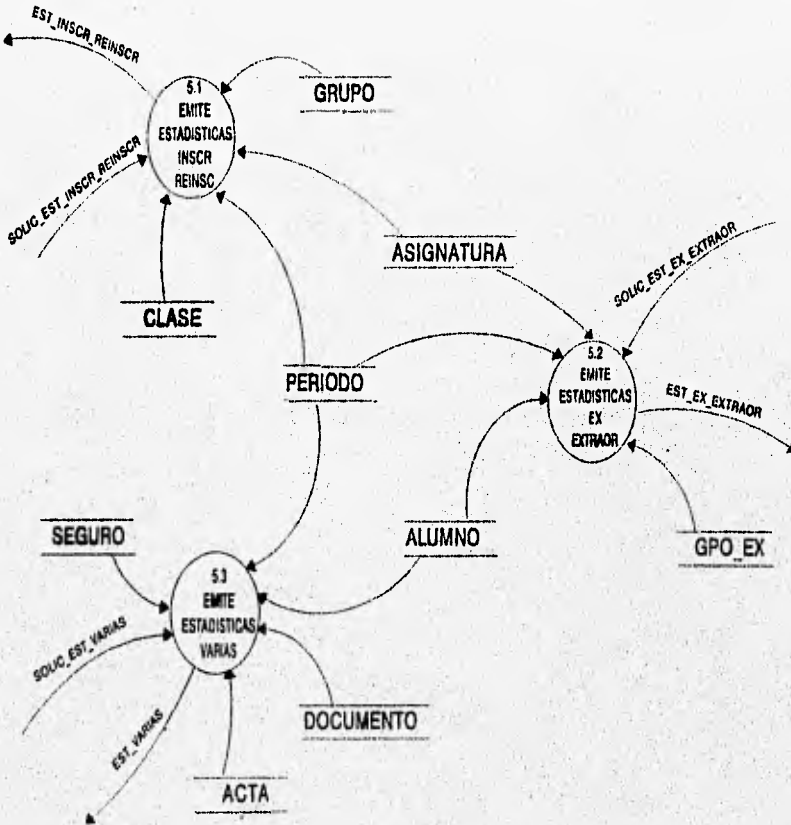
### D.F.D. 3 TRAMITA SEGURO FACULTATIVO



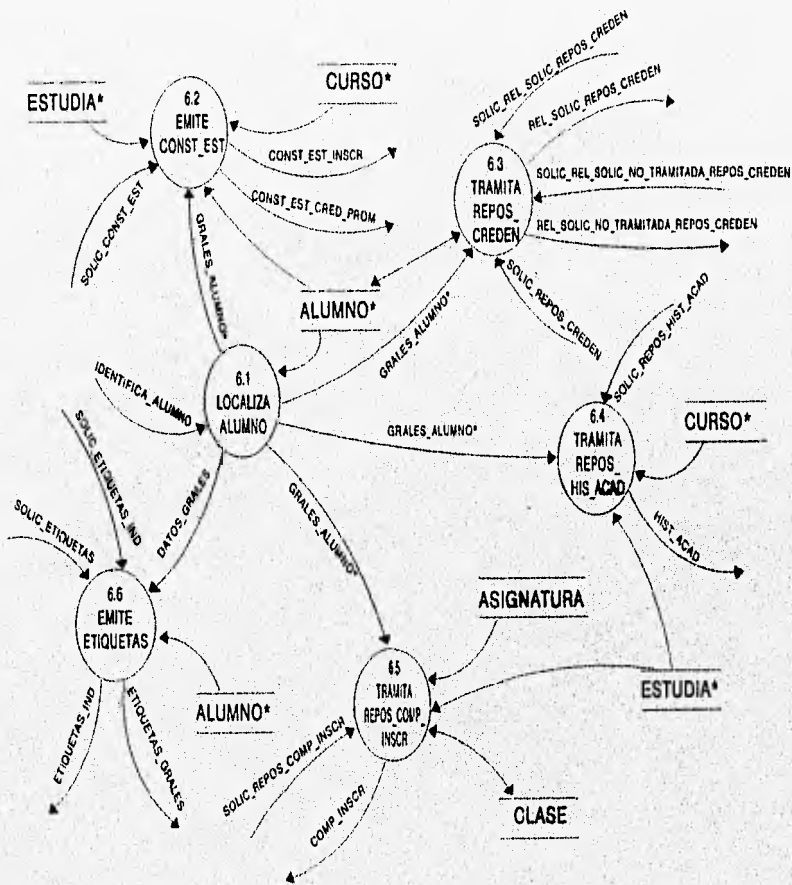
### D.F.D. 4 CONTROLA ACTAS



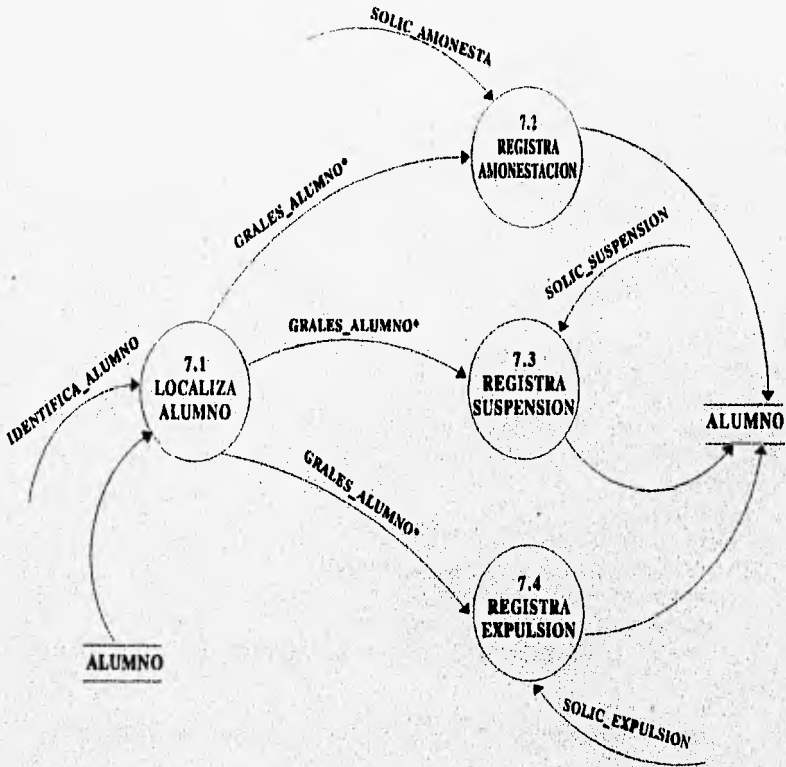
### D.F.D. 5 EMITE ESTADÍSTICAS



## D.F.D. 6 EMITE DOCUMENTOS



### D.F.D. 7 REGISTRA SANCCIONES ACADÉMICAS





### VI.2.2 Miniespecificaciones

Una miniespecificación es la descripción de que es lo que sucede en cada proceso primitivo de nivel más bajo en un diagrama de flujo de datos. Sin importar su nombre, el propósito de una miniespecificación es definir lo que debe hacerse para transformar entradas en salidas. [YOU93]

A manera de ejemplo, se presentan las siguientes miniespecificaciones del sistema.

#### 1.1.1, 2.1, 3.1 LOCALIZA ALUMNO

Obtén NO\_CTA y CVE\_PLANTEL  
 localiza en archivo alumno NO\_CTA y CVE\_PLANTEL  
 si existe despliega GRALES\_ALUMNO  
 sino despliega mensaje de error  
 fin si

#### 1.1.2 REGISTRA REINSCR

Lee GRUPOS\_REINSCR  
 obtén GRALES\_ALUMNO  
 hacer mientras haya NUMERO\_GRUPO y CVE\_ASIG  
 lee siguiente NUMERO\_GRUPO y CVE\_ASIG  
 busca NUMERO\_GRUPO y CVE\_ASIG en CLASE  
 si encontrado  
 si vacantes > 0  
 localiza en SERIACION las ASIGREQ  
 verifica si el alumno tiene aprobadas las ASIGREQ  
 si todas las ASIGREQ están aprobadas  
 escribe vacantes = vacantes - 1 en CLASE  
 escribe en CURSA NO\_CUENTA, NUMERO\_GRUPO, CVE\_PLANTEL,  
 CVEASIG  
 si no  
 manda mensaje de error "El alumno no ha aprobado asignaturas requisito"  
 fin si  
 si no  
 manda mensaje de error "El grupo no tiene cupo"  
 fin si  
 si no  
 manda mensaje de error "El grupo no existe"  
 end si  
 fin hacer  
 escribe en ESTUDIA inscrito = S, colegiatura=colegiatura de GRALES\_ALUMNO  
 genera DATOS\_INSCR

**1.1.3 REGISTRA INSCR**

```
Obtén GRALES_ALUMNO
lee GRUPO de PRIMER_INGRESO con NO_CUENTA=NO_CUENTA de
GRALES_ALUMNO
hacer mientras no sea fin de CLASE y GRUPO de CLASE= GRUPO de
PRIMER_INGRESO
  lee siguiente NUMERO_GRUPO y CVE_ASIG de CLASE
  si vacantes > 0
    valida seriación
    escribe vacantes = vacantes - 1 en CLASE
    escribe en CURSA NO_CUENTA, NUMERO_GRUPO, CVE_PLANTEL,
    CVEASIG
  si no
    manda mensaje de error "El grupo no tiene cupo"
  fin si
fin hacer
escribe en PRIMER_INGRESO INSCRITO = S, COLEGIATURA=COLEGIATURA
de GRALES_ALUMNO
genera DATOS_INSCR
```

**1.1.4 EMITE COMP\_BAJAS**

```
Obtén SOLIC_COMP_BAJA
Obtén DATOS_BAJA
imprime COMP_BAJAS
```

**1.1.5 EMITE COMP\_INSCR REGISTRO**

```
Obtén DATOS_INSCR
Busca NO_CUENTA de DATOS_INSCR en ALUMNO
si encontrado
  imprime COMP_INSCR
si no
  manda mensaje de error "Alumno inválido"
fin Si
```

**1.1.6 GENERA RELACION ALUMNOS REINSCR**

```
Obtén SOLIC_REL_ALUMNO_REINSCR
hacer mientras no sea fin de ESTUDIA
  leer siguiente ALUMNO
```

```
si inscrito = S
  imprime ALUMNO
fin Si
fin hacer
```

### 1.1.7 REGISTRA ALUMNO

Obtén DATOS\_PERS  
escribe o Actualiza en ALUMNO DATOS PERS

### 1.1.8 REGISTRA BAJA\_DEF\_VOL

```
Obtén SOLIC_BAJA_DEF_VOL
obtén GRALES_ALUMNO
obtén SOLIC_COMP_BAJA
imprime ALUMNO donde NO_CUENTA = NO_CUENTA DE GRALES_ALUMNO
borra ALUMNO donde NO_CUENTA = NO_CUENTA DE GRALES_ALUMNO
escribe ALUMNO en BAJASALUMNO
```

### 1.1.9 EMITE LISTADO\_BAJAS

```
Obtén SOLIC_REL_BAJA
hacer mientras no sea fin de BAJASALUMNO
  leer siguiente registro
  imprime registro
fin hacer
```

## 2.2 REGISTRA EX\_EXTRAOR

```
Si el alumno desea registrar hasta dos exámenes extraordinarios
  obtén CVEASIG
  localiza en el archivo SERIACION las ASIGREQ
  verifica si el alumno tiene aprobadas las ASIGREQ
  si todas las ASIGREQ están aprobadas
    registra ex_extraor
    genera DATOS_EXTRAOR
  si no
    manda mensaje de error "El alumno no ha aprobado
    asignaturas requisito"
  fin si
```

```
si no
  obtén CVEASIG
  localiza en el archivo SERIACION las ASIGREQ
  verifica si el alumno tiene aprobadas las ASIGREQ
  si todas las ASIGREQ están aprobadas
    registra el examen extraordinario
    imprime COMP_TRAMITE_EX_EXTRAOR
    solicita PERMISO_EXTRAOR
    si PERMISO_EXTRAOR_AUTOR = true
      imprime COMP_INSCR_EX_EXTRAOR
    si no
      no registra ex_extraor
    fin si
  fin si
fin si
```

### 2.3 AUTORIZA EX\_EXTRAOR

```
Despliega datos de PERMISO_EXTRAOR
lee historia académica del archivo ALUMNO
despliega historia académica
obtén autorización("S" o "N") y datos de PERMISO_EXTRAOR_AUTOR
guarda PERMISO_EXTRAOR_AUTOR en el archivo GPO_EX
```

### 2.4 REGISTRA CALENDARIO

```
Lee datos CALENDARIO
lee datos del archivo PROFESORES
si existen los datos en el archivo GPO_EX
  actualiza el archivo GPO_EX
sino
  escribe los datos leídos en el archivo GPO_EX
fin si
```

### 2.5 EMITE REPORTE DE EX\_EXTRAOR MAS\_2

```
Si recibes SOLIC_REP_ALUMNO_AUTO_MAS_2
  lee datos del archivo GPO_EX
  imprime ALUMNO_AUTO_MAS_2
fin si
```

**2.6 EMITE REPORTE DE ALUMNOS INSCRITOS**

```
Si recibes SOLIC_REP_ALUMNO_INSC_ASIG_JURADO
  lee datos del archivo GPO_EX
  imprime ALUMNO_INSC_ASIG_JURADO
fin si
```

**2.7 EMITE REPORTE DE PAGO A SINODALES**

```
Si recibes SOLIC_REP_PAGO_SINOD_EXTRAOR
  lee datos del archivo GPO_EX
  imprime REP_PAGO_SINOD_EXTRAOR
fin si
```

**2.8 EMITE COMP\_INSCR EXTRAOR**

```
Recibe DATOS_EXTRAOR
mientras NO_CUENTA ≠ NO_CUENTA en GPO_EX
  lee CVE_ASIG en GPO_EX
  imprime COMP_INSCR_EX_EXTRAOR
fin mientras
```

### VI.2.3 Diccionario de Datos

“Es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de datos y cálculos intermedios.” [YOU93]

El diccionario de datos define todos los elementos de los datos que se han declarado en los diagramas de flujo de datos a través de:

- La definición del significado de los flujos y archivos
- La especificación de la composición de los flujos y archivos.
- La especificación de las unidades y valores de los flujos y archivos.

El uso del diccionario de datos reduce ambigüedades y especifica detalles.

El significado debe explicar la relevancia del flujo de datos o del archivo del sistema. Asimismo, se deben evitar datos redundantes como la descripción de la composición de los datos o derivaciones, la repetición del significado del elemento ya que el elemento tiene su propia descripción y la repetición del nombre del dato.

Los símbolos siguientes se utilizan para definir la composición de un flujo de datos o de un archivo:

=	está compuesto de
+	y
()	optativo (puede estar presente)
{}	iteración
[]	seleccionar una de varias alternativas
**	comentario
	separa opciones alternativas en la construcción

[YOU93]

Idealmente cada término del diccionario de datos debe referirse a un elemento único en el modelo del sistema. Un alias es un nombre alternativo para algo ya definido en el diccionario.

**Diccionario de Datos**

acta\_complementaria\_extraordinario =  
\* acta complementaria de control de exámenes extraordinarios \*

acta\_complementaria\_ordinario =  
\* acta complementaria de control de exámenes ordinarios \*

acta\_extraordinario =  
\* acta de control de exámenes extraordinarios \*

acta\_ordinario =  
\* acta de control de exámenes ordinarios \*

alumno\_auto\_mas\_2 =  
1 {no\_cuenta + nomb\_alumno + num\_exámenes} N

alumno\_insc\_asig\_jurado =  
asignatura + cve\_jurado + 1 {no\_cuenta + nomb\_alumno } N

alumnos\_asig = 1 {dígito} 3

alumnos\_inscr = 1 {dígitos} 3

año\_esc =  
[01 | 02 | 03 | 04 | SDG]  
\*SDG = Sin Derecho a Grupo \*

año\_ing = 2 {dígito} 2

año\_ing\_ciclo = 4 {dígito} 4

ape\_materno\_alumno =  
\* apellido materno del alumno \*

ape\_materno\_profesor =  
\* apellido materno del profesor \*

ape\_paterno\_alumno =  
\* apellido paterno del alumno \*

ape\_paterno\_profesor =  
\* apellido paterno del profesor \*

asig\_a\_cursar = cve\_asig

asig\_aprob =  
\* Asignaturas aprobadas \*

asig\_cursada = cve\_asig

asig\_no\_aprob =  
\* Asignaturas no aprobadas \*

asignatura = nombre\_asig

asignatura = 1 {carácter}25

aula =  
\* espacio físico en que se dividen los edificios \*

autor = [si/no]

bajas = 1 {dígito}4

bajas\_art\_23 = 1 {dígito}4

calendario =  
cve\_asig + cve\_jurado + 1 {rfc\_sinodal + } 5 + lugar\_examen + fecha

calif =  
[MB | B | S | NA | NP | AC | RE | CO | 10 | 9 | 8 | 7 | 6 ]

calle =  
nombre\_calle + número

carta\_acepta =  
\* Documento en el cual se notifica al alumno que ha sido admitido en la UNAM \*

citatorio\_alumno =  
fecha + hora + lugar + nomb\_alumno

clínica =  
\* nombre de la institución donde el alumno recibirá servicios médicos \*

código\_postal = 5 {dígito} 5



colegiatura =  
 \* pago por parte del alumno, por concepto de reinscripción \*

colonia =  
 \* nombre de la colonia donde vive el alumno \*

comp\_bajas =  
 no\_cuenta + nomb\_alumno + fecha

comp\_inscr =  
 fecha\_emisión + num\_hoja + no\_cuenta + cve\_plantel + ape\_paterno\_alumno +  
 ape\_materno\_alumno + nombre\_alumno + año\_ing + periodo + página +  
 1 {cve\_plantel + nombre\_asig + créditos + año\_esc + número\_grupo} 7 +  
 firma\_alumno

comp\_inscr\_ex\_extraor =  
 {comp\_inscr\_ext\_normal | comp\_inscr\_ext\_2}

comp\_inscr\_ex\_obj =  
 folio + no\_cuenta + nomb\_alumno + cve\_plantel + fecha + hora + aula

comp\_inscr\_ext\_2 =  
 comp\_inscr\_ext\_normal

comp\_inscr\_ext\_normal =  
 1 {no\_cuenta + nomb\_alumno + número\_grupo + cve\_asig + asignatura +  
 jurado} N

comp\_prov =  
 comp\_insc + leyenda

comp\_seguro =  
 no\_cuenta + nomb\_alumno + fecha\_inicio + no\_afiliación + nombre\_escuela +  
 clínica

comp\_solicitud =  
 {comp\_inscr\_ex\_extraor | comp\_inscr | comp\_seguro | comp\_tranite\_ex\_extraor}

comp\_trámite\_ex\_extraord =  
 1 {no\_cuenta + nomb\_alumno + cve\_asig + asignatura + número\_grupo +  
 leyenda} N

comp\_trámite\_reg =  
[comp\_inscr]

const\_est\_cred\_prom =  
1{nomb\_alumno + no\_cuenta + plantel + año\_esc + prom + creditos\_cubiertos +  
horario + periodo + fecha}N

const\_est\_inscr =  
1{nomb\_alumno + no\_cuenta + plantel + año\_esc + horario + periodo + fecha}N

cred\_oblig =  
\* Créditos obligatorios \*

cred\_optat =  
\* Créditos optativos \*

créditos = 2 {dígito} 2

créditos\_cubiertos =  
\* suma de los créditos de cada asignatura aprobada \*

cuadro\_comp\_primer\_ing =  
1{año\_ing\_ciclo + plantel + cupos + alumnos\_asig + alumnos\_inscr}N

cupos = 1{dígitos}3

cve\_asig = 4 {dígito} 4

cve\_jurado = 4{dígito}4

cve\_plantel = 3 {dígito} 3

datos\_baja =  
grales\_alumno + fecha

datos\_extraor =  
grales\_alumno + fecha

datos\_inscr =  
grales\_alumno + fecha

datos\_pers =  
 ape\_paterno\_alumno + ape\_materno\_alumno + nomb\_alumno +  
 nacionalidad + fecha\_nac + edad + sexo + edo\_civil\_alumno + tel\_alumno  
 + medio\_transp + tiempo\_transp + escuela\_proced + tipo\_escuela\_proced  
 + promedio\_escuela\_proced + calle + colonia + delegación + entidad\_fed\_alumno +  
 código\_postal + empresa\_trabaja + padre\_alumno + seguro\_fac +  
 tel\_empresa\_trabaja + entidad\_fed\_empresa

datos\_seguro = clínica + no\_afiliación + lugar + vigencia + fecha\_admvo + fecha\_movto

delegación = 1{carácter}30

dependencias\_planteles = [biblioteca | secc\_escolar | secc\_admva | centro\_cómputo |  
 dirección ]

DGAE =  
 \* Dirección General de Administración Escolar \*

DGENP =  
 \* Dirección General de la Escuela Nacional Preparatoria \*

diario\_méxico =  
 \* entidad que publica las estadísticas de alumnos sobresalientes \*

doctos\_emite =  
 [etiquetas\_ind | etiquetas\_grales | const\_est\_inscr | const\_est\_cred\_prom |  
 hist\_acad | comp\_inscr]

domicilio =  
 calle + colonia + delegación + entidad\_fed\_alumno + código\_postal

edad = 1{dígito}2

edo\_civil =  
 [soltero | casado | viudo | divorciado | unión\_libre]

edo\_civil\_alumno = edo\_civil

egreso = 1{dígito}3

empresa\_trabaja =  
 \* en caso de que trabaje el alumno, nombre de la empresa \*

---

entidad\_fed\_alumno =  
 \* Nombre del Estado donde vive el alumno \*

entidad\_fed\_empresa =  
 \* en caso de que el alumno trabaje, Estado en el que se localiza la empresa \*

entregado = 1 { digito } 4

envia\_docts =  
 [ relación | estadísticas ]

escuela\_proced =  
 \* institución educativa, donde el alumno realizó sus estudios anteriores \*

est\_actas =  
 [ est\_actas\_grales | est\_actas\_rectif ]

est\_actas\_grales = 1 { periodo + plantel + tipo\_acta + total } N

est\_actas\_rectif = 1 { periodo + plantel + no\_actas + total } N

est\_alumnos\_diploma = 1 { periodo + plantel + generación + no\_lugar + total } N

est\_alumnos\_medalla = 1 { periodo + plantel + generación + num\_alumnos + promedio + total } N

est\_alumnos\_sobresalientes =  
 [ est\_alumnos\_diploma | est\_alumnos\_medalla ]

est\_asig\_primer\_ing = periodo + plantel + { número\_grupo + asignatura + nombre\_asig + total }

est\_bajas =  
 1 { periodo + plantel + mes\_tot + total }

est\_camb\_gpo\_bajas\_art\_23 =  
 1 { plantel + bajas\_art\_23 + camb\_gpo + total } N

est\_certificados =  
 1 { periodo + plantel + mes\_tot + total } N

est\_constancia\_100% =  
 1 { periodo + plantel + mes\_tot + total } N

---

est\_constancia\_sit\_esc =  
est\_constancia\_100%

est\_constancias =  
[est\_constancia\_100% | est\_constancia\_sit\_esc]

est\_credencial =  
[est\_tarjetón\_esp | est\_credenciales\_méxico]

est\_credenciales\_méxico =  
est\_tarjetón\_esp

est\_cuotas\_volunt =  
1{año\_esc + plantel + periodo + total}N

est\_cupos\_gpos = 1{periodo + plantel + año\_esc + número\_grupo + total}

est\_dictámenes =  
1{periodo + plantel + mes\_tot + total}N

est\_egresados =  
1{periodo + plantel + total}N

est\_ex\_extraor =  
[est\_ex\_extraor\_inscr\_asig | est\_ex\_extraor\_EB | est\_ex\_extraor\_inscr |  
est\_ex\_extraor\_EA | est\_ex\_extraor\_mas\_dos]

est\_ex\_extraor\_EA =  
est\_ex\_extraor\_EB

est\_ex\_extraor\_EB =  
1{periodo + plantel + inscritos + aprobados + no\_aprobados + total}N

est\_ex\_extraor\_inscr = 1  
{periodo + plantel + total}N

est\_ex\_extraor\_inscr\_asig =  
1{periodo + plantel + cve\_asig + asignatura + num\_jurado + total}N

est\_ex\_extraor\_mas\_dos =  
1{periodo + plantel + alumnos\_inscr + exam\_auto + total}N

est\_ex\_prof\_aprob\_rep =  
1{periodo + año + plantel + total}N

**est\_ex\_prof\_inscr =**  
 período + {plantel + num\_ext + total}

**est\_gpos\_asig =**  
 1 {período + plantel + número\_grupo + asignatura + total}N

**est\_inscr\_reinscr =**  
 [est\_asig\_primer\_ing | bajas\_art\_23 | cuadro\_comp\_primer\_ing |  
 est\_cuotas\_volunt | est\_cupos\_gpos | est\_gpos\_asig | est\_pago\_lab |  
 resumen\_inscr\_anual]

**est\_pago\_lab =**  
 1 {plantel + período + año\_esc + num\_alumnos + importe + total}N

**est\_rev\_estud =**  
 1 {período + plantel + tot\_tram\_unic + tot\_tram\_simul + tot\_certif + total}N

**est\_solic\_seg\_fac =**  
 1 {período + plantel + mes\_tot + bajas + entregado + no\_entregado + total}N

**est\_tarjetón\_esp =**  
 1 {período + plantel + mes\_tot + total}N

**est\_varias =**  
 [est\_actas | est\_constancias | est\_bajas | est\_certificados | est\_credencial |  
 est\_dictámenes | est\_alumnos\_sobresalientes | est\_solic\_seg\_fac]

**estadísticas =**  
 [est\_insc\_reinsc | est\_ex\_extraor | est\_varias]

**etiqueta\_ind =**  
 plantel + num\_comprobante + no\_cuenta + nomb\_alumno + tel + domicilio +  
 generación + colonia + código\_postal

**etiquetas\_grales =**  
 1 {plantel + num\_comprobante + no\_cuenta + nomb\_alumno + tel\_alumno +  
 domicilio + generación + colonia + código postal}N

**exam\_auto =** 1 {dígito}5

**extraor\_autor =**  
 nombre\_autor + autor + fecha\_movimiento

**extraor\_presenta =**  
 1 {cve\_asig + cve\_jurado} 6 + (solic\_mas\_2)

extraordinario =

\* examen que se presenta para acreditar una asignatura fuera del período ordinario \*

f = femenino

fecha =

día + mes + año

fecha\_admvo =

fecha \* día en el que el IMSS tramita el seguro facultativo \*

fecha\_emisión =

fecha \* día en que se emitió el documento \*

fecha\_ex = fecha

fecha\_fin\_sanción = fecha

fecha\_inici = fecha

fecha\_movimiento =

fecha \* día en que se autoriza la presentación del examen extraordinario,  
por parte del funcionario \*

fecha\_movto =

fecha \* día en el que la UNAM inicia trámite del seguro facultativo \*

fecha\_nac = día + mes + año

fecha\_sol =

fecha \* día en el que el alumno solicita el seguro facultativo \*

firma\_alumno = \* firma del alumno \*

folio = 1 {dígito}5

folio\_acta = 7 {dígito}7

folio\_rect = 7 {dígito}7

generación = año\_ing

gpo\_autor = \* Grupo autorizado \*

grado =  
 {licenciado | maestro | doctor | externo}

grales\_alumno =  
 nomb\_alumno | num\_comprobante + año\_ing + cve\_plantel

grupo\_actual = número\_grupo

grupo\_asig = cve\_asig

grupo\_autor = gpo\_autor

grupo\_deseado = número\_grupo

grupos\_reinscr =  
 pago\_lab + colegiatura + 1 {número\_grupo + cve\_asig} 7

hist\_acad =  
 fecha + nomb\_alumno + no\_cuenta + pagina + hoja + plantel + cve\_plantel  
 + año\_ing + plantel + cve\_plantel + periodo + cred\_oblig + cred\_optat +  
 asig\_apro + asig\_no\_apro + prom + 1 {cve\_asig + créditos + nombre\_asig +  
 calif + tipo\_exam + folio\_acta + número\_grupo + num\_ord + num\_ext} N

hoja =  
 \* número consecutivo asignado automáticamente \*

hora =  
 \* momento del día en que se realiza un evento determinado \*

horario =  
 \* periodo en el que se realiza una determinada actividad \*

identifica\_alumno =  
 no\_cuenta + cve\_plantel

identifica\_grupo =  
 cve\_plantel + número\_grupo + cve\_asig

importe =  
 1 {dígito}6 \* suma de las colegiaturas de todos los alumnos de un plantel,  
 un año\_esc y un periodo determinado \*

IMSS =  
 \* Instituto Mexicano del Seguro Social \*



ingreso = 1{dígito}3

inscritos = 1{dígito}6

leyenda =  
\* Comprobante provisional de trámite \*

listado\_bajas =  
1 {no\_cuenta + nomb\_alumno + plantel} N

lugar =  
\* espacio físico en donde se realiza una determinada actividad \*

lugar\_examen =  
fecha\_ex\_extraor + hora\_ex\_extraor + aula\_ex\_extraor

m = masculino

medio\_transp =  
[metro | camión | colectivo | particular | otro]

mes\_tot = 1{dígito}4

nacionalidad = [mex | ext]

no\_actas = 1{dígito}3

no\_actas\_rectifica = no\_actas

no\_afiliación = 1{dígito}17

no\_aprobado = \* dar por bueno el examen profesional \*

no\_aprobados = 1{dígito}6

no\_cuenta = 8 {dígito} 8

no\_entregado = 1{dígito}4

no\_lugar =  
[1o\_lugar | 2o\_lugar | 3er\_lugar]

no\_rep =  
\* Número de reposición de credencial \*

---

nomb\_alumno =  
ape\_paterno\_alumno + ape\_materno\_alumno + nombre\_alumno

nombre\_alumno =  
\* nombre de pila del alumno \*

nombre\_asig = 1 {carácter} 28

nombre\_autor =  
\* nombre del funcionario que autorizo que el alumno puede presentar el  
examen extraordinario \*

nombre\_calle = 1 {carácter} 35

nombre\_escuela = 1 {carácter} 30

nombre\_profesor = \* nombre de pila del profesor \*

num\_alumnos = 1 {dígito} 4

num\_comprobante = 1 {dígito} 5

num\_consecutivo = 1 {dígito} 3

num\_ex\_1 =  
1 {dígito} 3 \* número de exámenes con menos de 3 sinodales \*

num\_ex\_2 =  
1 {dígito} 3 \* número de exámenes con más de 2 sinodales \*

num\_exámenes = num\_ext

num\_ext =  
1 {dígito} 2 \* número de extraordinarios \*

num\_hoja = 1 {dígito} 3

num\_jurado = 1 {dígito} 2

num\_ord =  
1 {dígito} 1 \* número de ordinarios \*

número = 1 {dígito} 4

---

número\_grupo = 4 {dígito} 4

ordinario =

\* examen que el alumno presenta para acreditar una asignatura dentro del periodo establecido \*

padre\_alumno =

\* nombre del padre del alumno \*

página =

\* en caso de más de una página, número de página \*

pago\_lab = {si | no}

pago\_sinod\_extraor =

periodo + 1 {rfc\_sinodal + num\_examen1 + num\_exam2}N

periodo = 1 {dígito} 3

permiso\_extraor =

1 {no\_cuenta + nomb\_alumno + cve\_asig + cve\_jurado} 4

permiso\_extraor\_autor =

permiso\_extraor + 1 {extraor\_autor} 4

petición =

\* orden del operador de emitir un cierto reporte, comprobante, etc. \*

plantel =

{uno | dos | tres | cuatro | cinco | seis | siete | ocho | nueve}

profesor =

ape\_paterno\_profesor + ape\_materno\_profesor + nombre\_profesor

prom =

\* Promedio de las calificaciones \*

promedio = prom

promedio\_escuela\_proced =

\* representa el promedio del alumno, obtenido en la escuela de procedencia \*

rel\_actas =

1 {nomb\_alumno + plantel}N

rel\_actas\_faltantes =  
 periodo + fecha + 1 { asignatura + profesor + folio\_acta + número\_grupo +  
 tipo\_examen } N

rel\_actas\_para\_DGAE =  
 cve\_plantel + periodo + fecha + 1 { tipo\_acta + folio\_acta + asignatura + profesor  
 + número\_grupo + tipo\_examen } N

rel\_actas\_rectifica\_para\_cu =  
 plantel + periodo + [ folio\_acta + asignatura + profesor + número\_grupo +  
 tipo\_examen ]

rel\_alfab\_gral =  
 plantel + periodo + 1 { no\_cuenta + nomb\_alumno + año\_ing + } N  
 \* son todos los alumnos \*

rel\_alumno\_baja\_seguro =  
 periodo + fecha + 1 { no\_cuenta + ape\_paterno\_alumno + ape\_materno\_alumno +  
 nombre\_alumno }

rel\_alumno\_inscr =  
 1 { folio + no\_cuenta + nomb\_alumno + calificación } N

rel\_alumno\_reinscr =  
 1 { plantel + no\_cuenta + nomb\_alumno } N

rel\_alumno\_seguro =  
 periodo + fecha + 1 { no\_cuenta + ape\_paterno\_alumno + ape\_materno\_alumno +  
 nombre\_alumno }

rel\_bajas =  
 periodo + plantel + fecha + 1 { no\_cuenta + nombre\_alumno }

rel\_cont\_alumno =  
 [ rel\_alumno\_reinscr | comp\_bajas | rel\_bajas ]

rel\_folio\_asig =  
 cve\_plantel + periodo + 1 { folio\_acta + asignatura + profesor + número\_grupo +  
 tipo\_examen } N

rel\_grales\_alumno =  
 1 { nomb\_alumno | num\_comprobante + año\_ing + cve\_plantel } N

rel\_profesor\_asig =  
 cve\_plantel + periodo + 1 { profesor + asignatura + no\_actas\_rectifica } N

rel\_solic\_no\_tramitadas\_repos\_creden =  
período + plantel + I {no\_cuenta + nomb\_alumno}

rel\_solic\_repos\_creden =  
I {no\_cuenta + nomb\_alumno + plantel + no\_rep}N

rel\_solic\_rev\_estud =  
rel\_solic\_tipo\_rev

rel\_solic\_tipo\_rev =  
período + cve\_plantel + fecha + I {no\_cuenta + nomb\_alumno + tipo\_rev}N

rel\_solicitudes =  
no\_cuenta + nombre + plantel

rel\_trámite\_reg =  
[ rel\_cont\_alumno ]

relación =  
[rel\_trámite\_reg | rep\_ex\_extraor | rep\_seguro | rep\_control\_actas |  
rep\_control\_actas\_rectifica | rel\_actas | rep\_emisión\_doctos]

rep\_control\_actas =  
[rel\_actas\_para\_DGAE + rel\_actas\_faltantes]

rep\_control\_actas\_rectifica =  
[rel\_folio\_asig | rel\_actas\_rectifica\_para\_DGAE | rel\_profesor\_asig]

rep\_emisión\_doctos =  
[rel\_solic\_repos\_creden | rel\_solic\_no\_tramitadas\_repos\_creden]

rep\_ex\_extraor =  
[alumno\_auto\_mas\_2 | alumno\_insc\_asig\_jurado | rep\_pago\_sinod\_extraor]

rep\_pago\_sinod\_extraor =  
período + plantel + {rfc\_sinodal + profesor + num\_ex\_1 + num\_ex\_2 + total}

rep\_seguro =  
[rel\_alumno\_seguro | rel\_alumno\_baja\_seguro]

resultado =  
[aprobado | no\_aprobado]

resumen\_inscr\_anual =  
| { periodo + plantel + total } N

rfe\_sinodal =  
\* registro federal de contribuyentes del profesor sinodal \*

seguro\_fac =  
\* en caso de que el alumno tenga seguro facultativo, número de afiliación \*

sexo = [ m | f ]

solic\_acredita =  
número + nomb\_alumno + cve\_plantel + cve\_asig

solic\_act\_calif =  
identifica\_grupo + plantel + folio\_acta + calif

solic\_acta\_equival =  
número + no\_cuenta + calif + créditos

solic\_actas =  
[solic\_registro\_actas | solicit\_control\_actas | solicit\_control\_actas\_rectifica |  
solic\_registro\_actas\_rectifica]

solic\_alta\_seguro =  
identifica\_alumno + { fecha | solicit\_repos\_comp\_seguro }

solic\_amonesta = petición

solic\_auto\_mas\_2 = petición

solic\_baja\_def\_vol = petición

solic\_baja\_seguro =  
identifica\_alumno + fecha\_sol

solic\_citatorio\_alumno = petición

solic\_comp\_baja = petición

solic\_comp\_inscr = petición

solic\_comp\_inscr\_ex\_extraor = petición

solic\_const\_est = petición

solic\_control\_actas =  
tipo\_acta + folio\_acta

solic\_control\_actas\_rectifica =  
folio\_acta + no\_cuenta

solic\_cuadro\_comp\_primer\_ing = petición

solic\_def\_vol =  
no\_cuenta + nomb\_alumno

solic\_emisión\_doctos =  
identifica\_alumno + solicitación\_emisión\_doctos

solic\_emisión\_doctos =  
[solic\_etiquetas | solicitación\_etiquetas\_ind | solicitación\_const\_est |  
solic\_repos\_comp\_inscr | solicitación\_respos\_creden | solicitación\_respos\_hist\_acad]

solic\_est\_actas = petición

solic\_est\_alumnos\_sobresalientes = petición

solic\_est\_asig\_primer\_ing = petición

solic\_est\_bajas = petición

solic\_est\_certificados = petición

solic\_est\_constancias = petición

solic\_est\_credencial = petición

solic\_est\_cuotas\_volunt = petición

solic\_est\_cupos\_gpos = petición

solic\_est\_dictámenes = petición

solic\_est\_egresados = petición

solic\_est\_ex\_extraor =  
[solic\_est\_ex\_extraor\_inscr\_asig | solicitación\_est\_ex\_extraor\_EB |

solic\_est\_ex\_extraor\_inscr | solicit\_est\_ex\_extraor\_EA |  
solic\_est\_ex\_extraor\_mas\_dos]

solic\_est\_ex\_extraor\_inscr = petición

solic\_est\_ex\_extraor\_mas\_dos = petición

solic\_est\_ex\_extraor\_EA = petición

solic\_est\_ex\_extraor\_EB = petición

solic\_est\_ex\_extraor\_inscr\_asig = petición

solic\_est\_ex\_prof\_aprob\_rep = petición

solic\_est\_ex\_prof\_inscr = petición

solic\_est\_gpos\_asig = petición

solic\_est\_insc\_reinsc =

[solic\_est\_asig\_primer\_ing | solicit\_est\_camb\_gpo\_bajas\_art\_23 |  
solic\_est\_cuadro\_comp\_primer\_ing | solicit\_est\_cuotas\_volunt |  
solic\_est\_cuotas\_gpos | solicit\_est\_gpos\_asig | solicit\_est\_pago\_lab |  
solic\_est\_resumen\_inscr\_anual]

solic\_est\_pago\_lab = petición

solic\_est\_rev\_estud = petición

solic\_est\_solic\_seg\_fac = petición

solic\_est\_varias =

[solic\_est\_actas | solicit\_est\_bajas | solicit\_est\_constancias |  
solic\_est\_certificados | solicit\_est\_credencial | solicit\_est\_dictámenes |  
solic\_est\_alumnos\_sobresalientes | solicit\_est\_solic\_seg\_fac]

solic\_estadisticas =

[solic\_est\_insc\_reinsc | solicit\_est\_ex\_extraor | solicit\_est\_varias]

solic\_etiqueta\_ind = petición

solic\_etiquetas = petición



**solic\_ex\_EA** =  
 identifica\_alumno + 1 {extraor\_presenta} 6

**solic\_ex\_extraor** =  
 identifica\_alumno + extraor\_presenta

**solic\_expulsión** = petición

**solic\_ing\_años\_posteriores** = petición

**solic\_ing\_años\_poster** =  
 no\_cuenta + nomb\_alumno + cve\_plantel + año\_esc + 1 {cve\_asig + calif}N

**solic\_inscr** =  
 identifica\_alumno + datos\_pers

**solic\_listado\_bajas** = petición

**solic\_mas\_2** =  
 \* solicitud para presentar más de dos exámenes extraordinarios \*

**solic\_no\_tramitadas\_repos\_creden** =  
 no\_cuenta + nomb\_alumno + plantel + no\_rep

**solic\_plant** =  
 [solic\_actas | solicit\_estadísticas | solicit\_relación | solicit\_sanción ]

**solic\_prof** =  
 [solic\_actas\_calif | solicit\_relación | solicit\_sanción ]

**solic\_rect\_calif** =  
 identifica\_alumno + folio\_acta + folio\_rect + calif

**solic\_reg** =  
 [solic\_reinscr | solicit\_inscr ]

**solic\_registro\_actas** =  
 tipo\_acta + número\_grupo + cve\_asig + profesor + folio\_acta

**solic\_registro\_actas\_rectifica** =  
 profesor + folio\_acta + cve\_asig + tipo\_examen + año\_esc

**solic\_reinscr** =  
 identifica\_alumno + grupos\_reinscr + datos\_pers

solic\_rel\_actas = petición

solic\_rel\_actas\_faltantes = petición

solic\_rel\_actas\_para\_DGAE = petición

solic\_rel\_actas\_para\_DGENP = petición

solic\_rel\_actas\_rectifica\_para\_DGAE = petición

solic\_rel\_actas\_rectifica\_para\_DGENP = petición

solic\_rel\_alfab\_gral = petición

solic\_rel\_alumno\_baja\_seguro = petición

solic\_rel\_alumno\_inscr = petición

solic\_rel\_alumno\_reinscr = petición

solic\_rel\_alumno\_seguro = petición

solic\_rel\_bajas = petición

solic\_rel\_cont\_alumno =  
[solic\_rel\_alumno\_reinscr | solicit\_rel\_bajas | solicit\_comp\_baja | solicit\_comp\_inscr]

solic\_rel\_folio\_asig = petición

solic\_rel\_profesor\_asig = petición

solic\_rel\_solic\_no\_tramitadas\_repos\_creden = petición

solic\_rel\_solic\_repos\_creden = petición

solic\_rel\_solic\_rev\_estud = solicit\_rel\_solic\_tipo\_rev

solic\_rel\_solic\_tipo\_rev = tipo\_rev

solic\_rel\_solicitudes = petición

solic\_rel\_trámite\_reg =  
[solic\_rel\_cont\_alumno]

---

solic\_relación =  
[solic\_rel\_tramite\_reg | solie\_rep\_ex\_extraor | solie\_rep\_seguro |  
solic\_rep\_control\_actas | solie\_rep\_control\_actas\_rectifica | solie\_rel\_actas  
solic\_rep\_emisión\_doctos | solie\_rel\_alumno\_inscr ]

solic\_rep\_alumno\_auto\_mas\_2 = petición

solic\_rep\_alumno\_insc\_asig\_jurado = petición

solic\_rep\_control\_actas =  
[solic\_rel\_actas\_para\_DGAE | solie\_rel\_actas\_faltantes]

solic\_rep\_control\_actas\_rectifica =  
[solic\_rel\_folio\_asig | solie\_rel\_actas\_rectifica\_para\_DGAE |  
solic\_rel\_profesor\_asig]

solic\_rep\_emisión\_doctos =  
[solic\_rel\_solic\_repos\_creden | solie\_rel\_solic\_no\_tramitadas\_repos\_creden]

solic\_rep\_ex\_extraor =  
[solic\_rep\_alumno\_auto\_mas\_2 | solie\_rep\_alumno\_insc\_asig\_jurado |  
solic\_rep\_pago\_sinod\_extraor]

solic\_rep\_pago\_sinod\_extraor = petición

solic\_rep\_reinscr = petición

solic\_rep\_seguro =  
[solic\_rel\_alumno\_seguro | solie\_rel\_alumno\_baja\_seguro]

solic\_repos\_comp\_inscr = petición

solic\_repos\_comp\_seguro = petición

solic\_repos\_creden = petición

solic\_repos\_hist\_acad = petición

solic\_resumen\_inscr\_anual = petición

solic\_rev\_estud =  
identifica\_alumno + tipo\_rev

solic\_revalida =  
nomb\_alumno + cve\_plantel + cve\_asig

---

solic\_sancion =  
 identifica\_alumno + [solic\_amonesta | solic\_suspensión | solic\_expulsión]

solic\_suspensión =  
 fecha\_fin\_sanción + petición

solicitud =  
 [solic\_reg | solic\_ex\_extraor | solic\_baja\_seguro | solic\_alta\_seguro |  
 solic\_rect\_calif | solic\_rev\_estud | solic\_emisión\_doctos |  
 solic\_comp\_inscr\_ex\_extraor]

tarjetón\_cred =  
 \* Documento que se utiliza para llevar un control de las reposiciones de la  
 credencial del alumno \*

tel = teléfono

tel\_alumno = teléfono

tel\_empresa\_trabaja = teléfono

teléfono = 7 {dígito} 12

tiempo\_transp =  
 \* duración en horas en que el alumno tarda en trasladarse de su domicilio a  
 la escuela \*

tipo\_acta =  
 [acta\_ordinario | acta\_extraordinario | acta\_complementaria\_ordinario |  
 acta\_complementaria\_extraordinario]

tipo\_escuela\_proced =  
 [oficial | particular | extranjera]

tipo\_examen =  
 [ordinario | extraordinario]

tipo\_rev =  
 [trámite\_único | trámite\_certificado | trámite\_simultáneo]

tot\_certif = 1{dígito}3

tot\_tram\_simul = 1{dígito}3

tot\_tram\_unic = 1{dígito}3

total = 1 {dígito}6

trámite\_certificado =  
no\_cuenta + fecha\_sol

trámite\_simultáneo =  
no\_cuenta + fecha\_sol

trámite\_único =  
no\_cuenta + fecha\_sol

vigencia =  
\* tiempo de validez del seguro facultativo \*

### VL3 MODELO DE INFORMACIÓN

Una vez realizado el análisis de la información recopilada y de los requerimientos del usuario, el siguiente paso es identificar las entidades, así como las relaciones existentes entre éstas, y estructurarlas de tal forma que se obtenga un diagrama que representa el diseño lógico de la base de datos. Este diagrama es llamado Diagrama de Entidad - Relación (DER).

El principal propósito del DER es representar los objetos de datos y sus relaciones. Este diagrama consta de dos componentes principales:

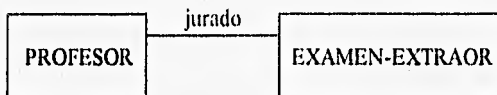
1. **Tipos de objetos.** Se representan por medio de un rectángulo en el diagrama. Son una colección o conjunto de objetos (cosas) del mundo real cuyos miembros juegan algún papel en el desarrollo del sistema; pueden además ser identificados de manera única y ser descritos por uno o más atributos.
2. **Relaciones.** Son la serie de conexiones o asociaciones entre los tipos de objetos que están conectados con la relación por medio de líneas.

Con la finalidad de facilitar la interpretación del Diagrama Entidad-Relación incluimos la explicación de la notación utilizada.

Las entidades se representan mediante un rectángulo dentro del cual se incluye el nombre de dicha entidad.

PROFESOR

Las relaciones se simbolizan con una línea que une a las entidades involucradas y se indica el nombre de la misma.



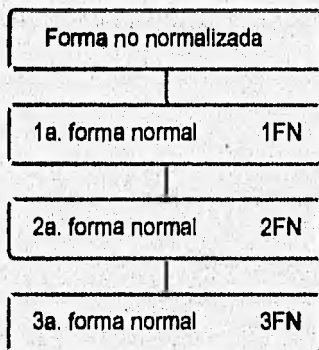
Las relaciones pueden ser entre dos ó más entidades, en el caso de sólo dos entidades la **cardinalidad** máxima se representa de la siguiente manera:

muchos a muchos	<u>M</u>	<u>N</u>
muchos a uno	<u>N</u>	<u>1</u>
uno a uno	<u>1</u>	<u>1</u>

Asimismo, es necesario realizar una descripción de las entidades, relaciones y los atributos de ambas.

Un segundo paso es aplicar la técnica de normalización, la cual, es una técnica desarrollada para asegurar que las estructuras de datos sean eficientes.

Los pasos para llevar a cabo la normalización son los siguientes:



---

Una relación está en la 1FN si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos. En este paso se convierte la estructura de datos a la forma de tablas bidimensionales.

Una relación está en la 2FN si es 1FN y cada atributo no llave de la relación es total y funcionalmente dependiente de su llave principal. En este paso cada atributo depende totalmente de la llave principal.

Una relación está en 3FN si es 2FN y ningún atributo no llave de la relación es funcionalmente dependiente de algún otro atributo no llave. En este paso se elimina cualquier dependencia transitiva de atributos no primos.

El atributo  $b$  de una relación  $r$  es **funcionalmente dependiente** del atributo  $a$  de la relación  $r$  si, en cada instante, cada valor de  $a$  está asociado con no más de un valor de  $b$  dentro de la relación  $r$ .

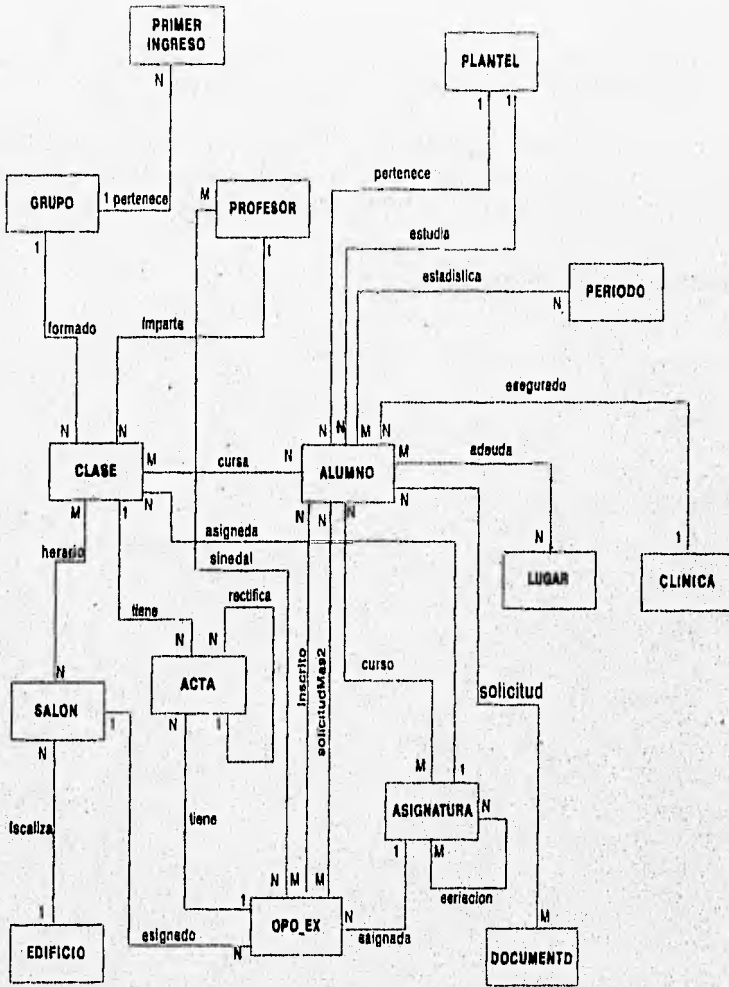
Suponga que  $a$ ,  $b$  y  $c$  son tres atributos de una relación  $r$ , si  $c$  es funcionalmente dependiente de  $b$  y  $b$  lo es de  $a$ , entonces  $c$  es funcionalmente dependiente de  $a$ . A esta dependencia se le conoce como **dependencia transitiva**.

Dentro de los beneficios que podemos encontrar con la aplicación de esta técnica tenemos:

- Nos libra de dependencias indeseables
- Se minimiza la reestructuración de datos
- Se mejora la dependencia de datos, permitiendo que las extensiones a la base de datos tengan pocos o ningún efecto sobre los programas o aplicaciones que la accesan.

En este trabajo sólo incluimos las entidades y relaciones ya **normalizadas**.

VI.3.1 Diagrama de Entidad Relación (DER)





---

## VI.4 ENTIDADES, RELACIONES Y ATRIBUTOS

A continuación mostramos la definición de las entidades y asociaciones presentadas en el DER. Únicamente presentamos aquellas relaciones para las cuales se obtiene una tabla en el Diseño Físico.

Para cada entidad y relación se listan sus atributos subrayando aquellos que formen parte de la llave primaria. Las llaves foráneas se indican en letra cursiva.

Para algunas entidades y relaciones, se definen algunos atributos duplicados o calculados, ya que se considera que beneficiarán el rendimiento de las consultas y operaciones que se lleven a cabo sobre la Base de Datos (redundancia necesaria).

### VI.4.1 Entidades y sus atributos

#### **acta**

*cveAsig*  
*cvePlantel*  
entregada  
folio  
*folioRec*  
*gpo*  
tipo  
rfe

#### **abamano**

colonia  
cp  
delegación  
domicilio  
edoCivil  
empresa  
entidad  
entidadProc  
fechaNac  
nacionalidad  
noCta  
nombre  
padre  
procedencia  
promedioProc  
sexo

tel  
telOfna  
tipoProc  
tiempoTrans  
transporte

**asignatura**

créditos  
cveAsig  
cvePlantel  
nombre  
añoEsc  
tipo  
anioPlan

**clase**

cupo  
cveAsig  
cvePlantel  
grupo  
vacantes

**clinica**

cveClinica  
nombre  
noAfiliación  
lugar  
vigencia  
fechaAdmvo  
fechaMovto

**documento**

cveDocto  
nombre

**edificio**

cveEdificio  
descripción

**grupo**

cvePlantel  
gpo

**grupoEx**

cveAsig  
cvePlantel  
fecha  
gpo  
noJurados  
periodo  
salón

**lugar**

cveLugar  
lugar

**periodo**

fechaFin  
fechaIni  
periodo

**plantel**

cvePlantel  
nombre

**primerIngreso**

colonia  
comprobante  
cp  
cuotaLab  
cuotaVol  
cvePlantel  
domicilio  
entidad  
gpo  
inscrito  
noCia  
nacionalidad  
nombre  
procedencia  
ingreso  
delegación  
edoCivil  
empresa  
entidadProc  
fechaNac  
padre  
promedioProc

sexo  
 tel  
 telOfna  
 tipoProc  
 transporte  
 tiempoTrans  
 anioPlan

**profesor**

nombre  
 dir  
rfc  
 tel  
 telOfna  
 grado

**salon**

salón  
 ubicación

**VI.4.2 Relaciones y sus Atributos**

**adeuda**

cveLugar  
noCta

**asegurado**

nombre  
 clinica  
 vigencia  
 noAfilación

**asignada**

cveAsig  
cvePlantel  
 gpo  
 añoEsc

**asignado**

salon  
cveAsig  
cvePlantel  
 periodo  
 fecha

**curso**

*cveAsig*  
*cvePlantel*  
*gpo*  
*noCta*

**curso**

calificación  
*cveAsig*  
*cvePlantel*  
folioActa  
gpo  
*noCta*  
*periodo*  
tipoExamen

**estadística**

La relación estadística en realidad no se compone de un conjunto de atributos, sino de un subconjunto de tablas con sus propios campos. Lo anterior es con el propósito de almacenar datos estadísticos del sistema. Cada tabla tiene su propia estructura, pero para este trabajo no consideramos indispensable incluirlas ya que sus campos de alguno u otra manera forman parte de todas las tablas que se describen en este capítulo.

\_estadística11  
\_estadística12  
\_estadística13  
\_estadística14  
\_estadística15  
\_estadística16  
\_estadística17  
\_estadística18  
\_estadística21  
\_estadística22  
\_estadística23  
\_estadística24  
\_estadística25  
\_estadística31  
\_estadística32  
\_estadística33  
\_estadística34  
\_estadística35  
\_estadística36  
\_estadística37  
\_estadística41  
\_estadística42

\_estadística43  
\_estadística46  
\_estadística51  
\_estadística52  
\_estadística53

**estudia**

aprobadas  
créditosRev  
cuotaLab  
cuotaVol  
cvePlantel  
ingreso  
inscrito  
noCta  
nombre  
promedio  
regAlumno  
reprobadas  
reposición  
últimoPeriodo  
añoEsc  
añoEscAnterior

**formado**

cveAsig  
gpo  
cupo

**horario**

cveAsig  
cvePlantel  
dia  
gpo  
hora  
salón

**imparte**

cveAsig  
cvePlantel  
gpo  
rfc

**inscrito**

cveAsig  
cvePlantel  
gpo  
periodo  
noCta

**localiza**

ubicación  
cveEdificio  
salon

**pertenece**

noCta  
cvePlantel  
ingreso  
grupo

**rectifica**

cveAsig  
folio  
folioRec  
fecha  
tipo

**solicitud**

cveDocto  
noCta  
fechaTram

**solicitudMas2**

autorizado  
cveAsig  
cvePlantel  
gpo  
noCta  
nomAutorizo  
periodo

**seriación**

cveAsig  
cveAsigReq  
cvePlantel  
anioPlan

tiene

*cveAsig*  
*cvePlantel*  
*entregada*  
*folio*  
*tipo*  
*grupo*

### VI.4.3 Descripción de las Entidades y sus Atributos

Para cada entidad se representa una descripción de ella, si se trata de una entidad dependiente, se indican las entidades de las que se depende y finalmente se describen cada uno de sus atributos.

Para cada atributo se representa una breve descripción indicando lo que representa dicho atributo y el tipo asociado. La cardinalidad requerida indica que el atributo no puede tener valor nulo.

#### **acta**

Documento emitido por la Dirección General de Administración Escolar (DGAE) de la Universidad Nacional Autónoma de México en donde el profesor de una asignatura o los sinodales de un examen extraordinario asientan las calificaciones de los alumnos.

#### **Descripción de los atributos:**

##### ***cveAsig***

Clave de la asignatura.

Atributo alfanumérico de cuatro dígitos.

Cardinalidad: Requerida.

##### ***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

##### ***entregada***

Atributo que indica si el acta ya fue entregada por el profesor o sinodal del grupo.

Atributo carácter que tiene los siguientes posibles valores:

N no ha sido entregada.

S ya fue entregada.



**folio**

Número de folio del acta en que se asienta la calificación. Cada hoja representa un acta con diferente folio.

Atributo numérico de 7 dígitos.

Cardinalidad: Requerida.

**folioRec**

Si el acta es de rectificación, este atributo hace referencia al folio del acta que se rectifica.

Atributo numérico de 7 dígitos.

**gpo**

Grupo en donde se imparte la asignatura o grupo de extraordinario.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

**tipo**

Este atributo indica si el acta es de rectificación o normal.

Atributo carácter, con los siguientes posibles valores:

N	acta normal.
R	acta de rectificación.

**rfc**

Registro Federal de Contribuyentes del profesor.

Atributo alfanumérico de 13 dígitos.

**alumno**

Estudiante de la Escuela Nacional Preparatorio, que fue aceptado para estudiar en alguno de los nueve planteles con que cuenta esta Institución que forma parte de la Universidad Nacional Autónoma de México.

**Descripción de los atributos:****colonia**

Para alumnos que radican en el D.F., colonia de residencia. Para alumnos del interior de la República Mexicana, es el nombre de la población o ciudad.

Atributo alfanumérico de 30 caracteres.

**cp**

Código Postal del domicilio del alumno.

Atributo alfanumérico de 5 dígitos.

**delegación**

Para alumnos que radican en el D.F., código de la delegación política en donde se encuentra su domicilio. Para alumnos del interior de la república Mexicana, es el código del municipio. Atributo numérico de 2 dígitos.

**domicilio**

Lugar de residencia del alumno, está formado por el nombre de la calle, el número exterior y el número interior.

Atributo alfanumérico de 30 caracteres.

**edoCivil**

Estado civil del alumno.

Atributo carácter con el siguiente significado:

C	casado
D	divorciado
S	soltero
V	viudo
O	otro

**empresa**

Si el alumno trabaja, nombre legal de la Empresa en donde trabaja.

Atributo alfanumérico de 30 caracteres.

**entidad**

Código de la entidad federativa en donde se encuentra el domicilio del alumno.

Atributo numérico de 2 dígitos.

**entidadProc**

Código de la entidad federativa de la escuela de procedencia del alumno.

Atributo numérico de 2 dígitos.

**fechaNac**

Fecha de nacimiento del alumno.

Atributo tipo fecha

**nacionalidad**

Nacionalidad del alumno.

Atributo carácter con los siguientes posibles valores:

M	mexicano
E	extranjero

**noCta**

Número de Cuenta, identificación de un alumno registrado en la UNAM. El número de Cuenta está formado por 8 dígitos con el siguiente formato:

AAFFFFV

donde:

AA	es el año lectivo en el que el alumno ingreso a la UNAM.
FFFFF	es el número de folio en el registro del periodo.
V	es un digito verificador atributo alfanumérico.

Cardinalidad: Requerida.

***nombre***

Nombre con el que el alumno queda registrado en la UNAM.  
Atributo alfanumérico de 32 caracteres.

Cardinalidad: Requerida

***padre***

Nombre del padre o tutor del alumno  
Atributo alfanumérico de 32 caracteres.

***procedencia***

Código de la escuela de procedencia del alumno.  
Atributo alfanumérico de 2 dígitos.

***promedioProc***

Promedio obtenido por el alumno en su escuela de procedencia.  
Atributo numérico real.

***sexo***

Sexo del alumno.  
Atributo de tipo carácter:

M	masculino
F	femenino
blanco	no proporcionado

***tel***

Teléfono del domicilio del alumno.  
Atributo alfanumérico de 10 caracteres.

***telOfna***

Si el alumno trabaja, teléfono de la empresa en donde se le puede localizar.  
Atributo alfanumérico de 10 caracteres.

***tipoProc***

Carácter de la escuela de procedencia del alumno, puede ser:

O	oficial
P	particular
N	otro.

***tiempoTrans***

Tiempo que el alumno tarda en llegar de su casa a la escuela.

Atributo alfanumérico de 2 dígitos.

***transporte***

Medio principal de transporte que el alumno utiliza para llegar al plantel de la ENP donde estudia.

Atributo de 4 caracteres los cuales pueden tener los siguientes valores.

M	metro
C	colectivo
P	particular
O	otro.

**asignatura**

Asignatura que está o estuvo contemplada en el plan de estudios de la ENP.

**Descripción de atributos:*****créditos***

Valor académico de una asignatura, que representa el cursarla satisfactoriamente, de acuerdo a los planes de estudio vigentes.

Atributo numérico de 2 dígitos.

Cardinalidad: Requerida.

***cveAsig***

Clave única de una asignatura en la ENP.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***nombre***

Nombre de la asignatura.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***añoEsc***

Año escolar al que pertenece la asignatura dentro del plan correspondiente.

Atributo numérico de 2 dígitos.

Cardinalidad: Requerida.

**tipo**

Este atributo indica si la asignatura está catalogada optativa u obligatoria, sus valores pueden ser:

O	Optativa
B	Obligatoria

Atributo de tipo carácter.

**anioPlan**

Año a partir del cual entro en vigor el plan de estudios.

Atributo numérico de 4 dígitos.

**clase**

Clase a la cual los alumnos se inscriben para cursar una asignatura, dentro de un grupo.

**Descripción de los atributos:****cupo**

Cupo estimado para la clase.

Atributo numérico de 3 dígitos.

**cveAsig**

Clave de la asignatura correspondiente.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

**cvePlantel**

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

**grupo**

Grupo al que pertenece la clase. Un grupo formado por las asignaturas correspondientes a un semestre de una carrera.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

**vacantes**

Este atributo indica el número de vacantes en la clase. Inicialmente tiene un valor igual al atributo *cupo*.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida

**clínica**

Dependencia del IMSS que otorga los servicios de Seguro Facultativo al alumno de la UNAM que se encuentra asegurado.

**Descripción de atributos:*****cveClinica***

Clave que se le asigna a cada una de las clínicas del IMSS

Atributo numérico de 2 dígitos.

Cardinalidad: Requerida

***nombre***

Nombre con el que el alumno queda registrado en la UNAM.

Atributo alfanumérico de 32 caracteres.

Cardinalidad: Requerida

***no.Afiliación***

Número con el que el asegurado queda registrado

Atributo numérico de 17 dígitos

Cardinalidad: Requerida

***lugar***

Espacio físico donde se ubica la clínica

Atributo alfanumérico de 40 caracteres

Cardinalidad: Requerida

***vigencia***

Tiempo de validez del seguro facultativo

Atributo tipo fecha

Cardinalidad: Requerida

***fechaAdmvo***

Día en el que el IMSS tramita el seguro facultativo

Atributo tipo fecha

***fechaMovto***

Día en el que la UNAM inicia trámite del seguro facultativo

Atributo tipo fecha.

**documento**

Esta entidad representa a los diferentes documentos que un alumno puede solicitar en Administración Escolar para algún propósito. En el sistema únicamente se necesita saber quien los solicita en el período actual y cuantos se solicitaron en períodos anteriores.

**Descripción de atributos:*****cveDocto***

Identificador único asignado por el sistema.

Atributo numérico secuencial.

Cardinalidad: Requerida.

***nombre***

Nombre del documento. Este puede ser:

- Carta de buena conducta
- Certificado de estudios parciales
- Constancia de Estudios
- Constancia de situación escolar
- Constancia para Servicio Médico
- Solicitud de credencial interna
- Solicitud de baja definitiva voluntaria
- Solicitud de inscripción al seguro facultativo del IMSS
- Solicitud de revisión de estudios
- Solicitud de Historial Académico

Atributo alfanumérico de 60 caracteres.

Cardinalidad: Requerida.

**edificio**

Edificio o lugar físico donde puedan existir salones, laboratorios, aulas o un espacio destinado a la impartición de clases de alguna asignatura de la ENP.

**Descripción de atributos:*****cveEdificio***

Identificador único asignado por el sistema.

Atributo numérico entero secuencial.

Cardinalidad: Requerida.

***descripción***

Nombre o descripción del edificio.

Atributo alfanumérico de 40 caracteres.

Cardinalidad: Requerida.

**grupo**

La entidad grupo no hace referencia a la impartición de una asignatura en un horario, sino más bien un conjunto de clases correspondientes a las asignaturas de un semestre de una carrera.

**Descripción de los atributos:*****cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***gpo***

Identificador de grupo.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

**grupoEx**

Grupo que se abre para inscripción a examen extraordinario en el periodo escolar actual. Los grupos de examen extraordinario pueden ser de tres tipos:

*Examen extraordinario EA (primer periodo) y examen extraordinario EB (segundo periodo).*

Es el examen al que tienen derecho los alumnos cuando:

- Habiéndose inscrito en la asignatura, no hayan llenado los requisitos para acreditarla.
- Siendo alumnos, no hayan estado inscritos en la asignatura correspondiente, o no la hayan cursado.
- Habiendo estado inscritos dos veces en una asignatura, no pueden inscribirse nuevamente.
- Hayan llegado al límite de tiempo en que pueden estar inscritos en la Universidad.

*Examen extraordinario EZ (examen especial, en un tercer periodo)*

Este examen especial sólo lo pueden realizar alumnos que hayan estado inscritos por última vez en el tercer año del bachillerato.

**Descripción de atributos:*****cveAsig***

Clave única de una asignatura en una carrera.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.



***fecha***

Fecha en la que se realizará el examen extraordinario. Incluye día y hora.

Atributo tipo fecha.

***gpo***

Identificador de grupo. Es un identificador alfanumérico de cuatro caracteres.

Cardinalidad: requerida

***noJurados***

Atributo que indica el número de sinodales para el examen.

Cálculo: número de renglones en la tabla de relación *sinodal* pertenecientes a este examen extraordinario.

Atributo numérico de 1 dígito.

***periodo***

Período de exámenes extraordinarios en el semestre.

Atributo alfanumérico de 3 caracteres.

Cardinalidad: Requerida.

***salón***

Identificador de un salón en donde se pueden impartir asignaturas o aplicar exámenes extraordinarios.

Atributo alfanumérico de 6 caracteres.

**lugar**

Departamento, laboratorio, biblioteca, o lugar en el que se puede dar un servicio de préstamo de material, y en el que el alumno se puede encontrar como deudor de material.

**Descripción de atributos:*****cveLugar***

Identificador único asignado por el sistema.

Atributo numérico secuencial.

Cardinalidad: requerida.

***lugar***

Nombre del lugar en donde se da el servicio de préstamo de material.

Atributo alfanumérico de 40 caracteres.

Cardinalidad: Requerida.

**periodo**

Período escolar en el cual se cursan asignaturas de las carreras que se imparten en la ENP.

La duración de un período escolar es de un año escolar.

**Descripción de atributos:*****fechaFin***

Fecha en la que termina oficialmente el período escolar.

Atributo de tipo fecha.

Cardinalidad: Requerida.

***fechaIni***

Fecha en la que comienza oficialmente el período escolar.

Atributo de tipo fecha.

Cardinalidad: Requerida.

***período***

Identificador del período escolar. Con el siguiente formato:

AAN

donde:

AA

últimos dos dígitos del año.

N

Período anual, este dígito no aplica y vale cero. En caso de exámenes extraordinarios valdrá 1, 2 y 3 (período especial)

Cardinalidad: Requerida.

**plantel**

Escuela de la Dirección General de la Escuela Nacional Preparatorias (UNAM) donde se imparten asignaturas a nivel bachillerato o técnico.

**Descripción de atributos:*****cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***nombre***

Nombre con el que se designa al plantel en la UNAM.

Atributo alfanumérico de 60 caracteres.

Cardinalidad: Requerida.

**primerIngreso**

Estudiante de primer ingreso a la Escuela Nacional Preparatoria (UNAM), que fué aceptado para estudiar el bachillerato en alguno de sus planteles.

---

**Descripción de atributos:*****colonia***

Para alumnos que radican en el D.F., colonia de residencia. Para alumnos del interior de la República Mexicana, es el nombre de la población o ciudad.

Atributo alfanumérico de 30 caracteres.

***comprobante***

Número del comprobante de aceptación que el alumno recibe en su domicilio.

Atributo numérico.

Cardinalidad: Requerida.

***cp***

Código Postal del domicilio del alumno.

atributo alfanumérico de 5 dígitos.

***cuotaLab***

Cuota que el alumno aporta al plantel de la ENP por concepto de laboratorio.

Atributo numérico real.

***cuotaVol***

Cuota que el alumno aporta en forma voluntaria por concepto de su inscripción a la ENP.

Atributo numérico real.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***delegación***

Para alumnos que radican en el D.F., código de la delegación política en donde se encuentra su domicilio. para alumnos del interior de la República Mexicana, es el código del municipio.

Atributo numérico de 2 dígitos.

***domicilio***

Lugar de residencia del alumno, está formado por el nombre de la calle, el número exterior y el número interior.

Atributo alfanumérico de 20 caracteres.

***entidad***

Código de la entidad federativa en donde se encuentra el domicilio del alumno.

Atributo numérico de 2 dígitos.

***gpo***

Identificador del grupo de primer año en el que está inscrito el alumno.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

***inscrito***

Atributo que indica si el alumno terminó los trámites de su inscripción en el plantel que se le asignó de la ENP. Si el alumno no lleva a cabo estos trámites, posteriormente será dado de baja en DGAE.

Atributo carácter con los siguientes posibles valores:

- |   |                           |
|---|---------------------------|
| S | si concluyó sus trámites. |
| N | no concluyó sus trámites. |

***nacionalidad***

Nacionalidad del alumno.

Atributo carácter con los siguientes posibles valores:

- |   |             |
|---|-------------|
| M | mexicano.   |
| E | extranjero. |

***noCta***

Número de Cuenta, Identificación de un alumno registrado en la UNAM.

Atributo alfanumérico de 8 caracteres.

Cardinalidad: Requerida.

***ingreso***

Año de ingreso del alumno a la ENP.

Atributo numérico de 4 dígitos.

Cardinalidad: Requerida.

***estadoCivil***

Estado civil del alumno.

Atributo carácter con el siguiente significado:

- |   |             |
|---|-------------|
| C | casado.     |
| D | divorciado. |
| S | soltero.    |
| V | viudo.      |
| O | otro.       |

***empresa***

Si el alumno trabaja, nombre legal de la Empresa en donde trabaja.

Atributo alfanumérico de 30 caracteres.

**entidadProc**

Código de la entidad federativa de la escuela de procedencia del alumno.  
Atributo numérico de 2 dígitos.

**fechaNac**

Fecha de nacimiento del alumno.  
Atributo tipo fecha.

**padre**

Nombre del padre o tutor del alumno.  
Atributo alfanumérico de 32 caracteres.

**promedioProc**

Promedio obtenido por el alumno en su escuela de procedencia.  
Atributo numérico real.

**sexo**

Sexo del alumno.  
Atributo de tipo carácter:

M	masculino.
F	femenino
blanco	no proporcionado

**tel**

Teléfono del domicilio del alumno.  
Atributo alfanumérico de 10 caracteres.

**telOfina**

Si el alumno trabaja, teléfono de la empresa en donde se le puede localizar.  
Atributo alfanumérico de 10 caracteres.

**tipoProc**

Carácter de la escuela de procedencia del alumno, puede ser:

O	oficial
P	particular
N	otro.

**tiempoTrans**

Tiempo que el alumno tarda en llegar de su casa a la escuela.  
Atributo alfanumérico de 2 dígitos.

**transporte**

Medio principal de transporte que el alumno utiliza para llegar al plantel de la ENP donde estudia.

Atributo de 4 caracteres los cuales pueden tener los siguientes valores.

M	metro
C	colectivo
P	particular
O	otro.

### ***anioPlan***

Año a partir del cuál entró en vigor el plan de estudios.

Atributo numérico de 4 dígitos.

### **profesor**

Personal de la ENP contratado en alguna plaza académica como profesor.

#### **Descripción de atributos:**

#### ***nombre***

Nombre del profesor.

Atributo alfanumérico de 32 caracteres.

Cardinalidad: Requerida.

#### ***dir***

Lugar de residencia del profesor, formado por calle, número y colonia.

Atributo alfanumérico de 40 caracteres.

#### ***rfc***

Registro Federal de Causantes del profesor.

Atributo alfanumérico de 13 caracteres con el siguiente formato:

**ABCDAAAMDDHHH**

donde:

A	primer letra del primer apellido.
B	primer vocal del primer apellido.
C	primer letra del segundo apellido.
D	primer letra del primer nombre.
AA	dos últimos dígitos del año de nacimiento.
DD	día del mes de nacimiento.
HHH	homoclave.

Cardinalidad: Requerida

#### ***tel***

Teléfono del domicilio del profesor.

Atributo alfanumérico de 10 caracteres.

**telOfna**

Teléfono del lugar de trabajo del profesor.

Atributo alfanumérico de 10 caracteres.

**grado**

Nivel de estudios del profesor.

Atributo alfanumérico de 7 caracteres.

**salón**

Salón, aula, sala, laboratorio o lugar en donde se pueden impartir asignaturas de las asignaturas de la ENP. o aplicar exámenes extraordinarios.

**Descripción de atributos:****salón**

Identificador del salón.

Atributo alfanumérico de 6 caracteres.

Cardinalidad: Requerida.

**ubicación**

Identificador del edificio o lugar de la ubicación física del salón.

Atributo numérico de 2 dígitos.

**VI.4.4 Descripción de las Relaciones y sus Atributos**

Esta sección complementa la anterior, detallando cada una de las características de las relaciones generadas del DER (Diagrama Entidad Relación), así como sus atributos. Solamente se presentan las relaciones para las cuales se generará tabla de Disco Físico.

Para cada relación se presenta una descripción de ella, la cardinalidad requerida y finalmente se describen cada uno de sus atributos.

Para cada atributo se presenta una breve descripción indicando su longitud y el tipo asociado. La cardinalidad Requerida indica que el atributo no puede tener valor nulo.

**adeuda**

Indica los lugares en donde un alumno puede tener un adeudo.

**Descripción de atributos:*****cveLugar***

Clave del lugar de adeudo.

Atributo numérico de 2 dígitos.

Cardinalidad: Requerida.

***noCta***

Número de cuenta del alumno que tiene el adeudo.

Atributo alfanumérico de 8 dígitos.

Cardinalidad: Requerida.

**asignada**

Información que mantiene la relación entre las asignaturas de una clase y con grupo\_ex

**Descripción de atributos:*****cveAsig***

Clave de la asignatura que se asigna a una clase o a un gpo\_ex

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***gpo***

Grupo en el que se cursa la asignatura.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

***añoEsc***

Es el año escolar en el que el alumno se encuentra inscrito.

Atributo numérico de 2 dígitos.

**asignado**

Mantiene la relación de cada salón con la entidad grupo\_ex.



**Descripción de atributos:****salón**

Identificador del salón que se asigna a grupo\_ex.  
Atributo alfanumérico de 6 caracteres.

**cveAsig**

Clave de la asignatura que se asigna a una clase o a un gpo\_ex  
Atributo alfanumérico de 4 dígitos.  
Cardinalidad: Requerida.

**cvePlantel**

Clave del Plantel a la que pertenece la asignatura.  
Atributo numérico de 3 dígitos.  
Cardinalidad: Requerida.

**periodo**

Periodo escolar correspondiente.  
Atributo alfanumérico de 3 caracteres.  
Cardinalidad: Requerida.

**fecha**

Fecha en la que se realizará el examen extraordinario. Incluye día y hora.  
Atributo tipo fecha.

**seguro**

Relación que mantiene la información de los alumnos que cuentan con seguro facultativo.

**Descripción de atributos:****nombre**

Nombre del alumno que obtiene el seguro facultativo.  
Atributo alfanumérico de 32 caracteres.  
Cardinalidad: Requerida.

**cveClinica**

Clave que se le asigna a cada una de las clínicas del IMSS  
Atributo numérico de 2 dígitos.  
Cardinalidad: Requerida

**vigencia**

Tiempo de validez del seguro facultativo  
Atributo tipo fecha  
Cardinalidad: Requerida

***no:Afiliación***

Número con el que el asegurado queda registrado

Atributo numérico de 17 dígitos

Cardinalidad: Requerida

**formado**

Indica la información de las clases que cursan actualmente los alumnos en un grupo.

**Descripción de atributos:*****cveAsig***

Clave de la asignatura que actualmente cursa el alumno.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***gpo***

Grupo en el que se cursa la asignatura.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

***cupo***

Cupo estimado para la clase.

Atributo numérico de 3 dígitos.

**CURSA**

Esta relación mantiene la información de las asignaturas que cursan actualmente los alumnos.

**Descripción de atributos:*****cveAsig***

Clave de la asignatura que actualmente cursa el alumno.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***gpo***

Grupo en el que se cursa la asignatura.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

***noCta***

Número de cuenta del alumno que cursa la asignatura.

Atributo alfanumérico de 8 dígitos.

Cardinalidad: Requerida.

**curso**

Esta relación mantiene la información de las asignaturas que ha presentado el alumno, ya sea en examen ordinario o extraordinario. esta relación refleja la Historia Académica del alumno en un plantel de la ENP.

**Descripción de atributos:*****calificación***

Código que representa la última calificación obtenida por un alumno al presentar un examen ordinario o extraordinario de una asignatura.

Atributo alfanumérico de 2 caracteres, con los siguientes posibles valores:

0 - 10	calificación numérica
MB	muy bien
B	bien
S	suficiente
NA	no acreditada
NP	no presentada
AC	acreditada
RE	revalidada
CO	covalidada

Cardinalidad: Requerida

***cveAsig***

Clave de la asignatura que el alumno ha presentado de alguna forma (ordinario, extraordinario o ambas).

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***folioActa***

folio del acta en donde se acentó la última calificación de la asignatura.

Atributo numérico de 7 dígitos.

Cardinalidad: Requerida.

***gpo***

Grupo en el que se presentó por última vez la asignatura.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

***noCta***

Número de cuenta del alumno que cursó la asignatura.

Atributo alfanumérico de 8 dígitos.

Cardinalidad: Requerida.

***periodo***

Período escolar en el que se presentó la asignatura.

Atributo alfanumérico de 3 caracteres.

Cardinalidad: Requerida.

***tipoExamen***

Tipo de examen en donde se presentó la asignatura.

Atributo carácter con los siguientes posibles valores:

Z	Extraordinario de tipo especial
E	Extraordinario
O	Ordinario.

Cardinalidad: Requerida.

***estadística***

Esta relación mantiene la información referente a las estadísticas anuales que se obtienen referentes a varios tópicos.

***estudia***

Esta relación mantiene la información referente a las asignaturas de un alumno. La mayor parte de los atributos son calculados o duplicados, esto con el propósito de obtener rápidamente parámetros que determinan el avance del alumno.

**Descripción de atributos:*****aprobadas***

Número de asignaturas que han sido acreditadas de alguna forma (ordinario, extraordinario, revalidación, etc.).

Este atributo se obtiene sumando todas las asignaturas acreditadas en la relación *curso*.

Atributo numérico de 2 dígitos.

Cardinalidad: Requerida.

***creditosOblig***

Número de créditos que han aportado las asignaturas obligatorias del plan de estudios que han sido acreditadas mediante examen ordinario o extraordinario.

Este atributo se obtiene sumando los créditos de todas las asignaturas obligatorias acreditadas registradas en la relación *curso*.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***creditosOpta***

Número de créditos que han aportado las asignaturas optativas del plan de estudios que han sido acreditadas mediante examen ordinario o extraordinario.

Este atributo se obtiene sumando los créditos de todas las asignaturas optativas acreditadas registradas en la relación *curso*.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***cuotaLab***

Cuota que el alumno aporta al plantel de la ENP por concepto de laboratorio.

Atributo numérico real.

***cuotaVol***

Cuota que el alumno aporta en forma voluntaria por concepto de su inscripción a la ENP.

Atributo numérico real.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***ingreso***

Año de ingreso del alumno a la ENP.

Atributo numérico de 4 dígitos.

Cardinalidad: Requerida.

***inscrito***

Atributo que indica si el alumno está inscrito, en el periodo actual ordinario, en alguna de las asignaturas correspondientes a la carrera que cursa.

Atributo carácter que tiene los siguientes posibles valores:

0	no inscrito.
1	inscrito (registrado en la relación <i>inscrito</i> ).

***noCta***

Número de cuenta del alumno que cursó la asignatura.

Atributo alfanumérico de 8 dígitos.

Cardinalidad: Requerida.

***nombre***

Nombre del alumno que cursa el bachillerato.

Atributo alfanumérico de 32 caracteres.

Cardinalidad: Requerida.

***promedio***

Atributo que indica el promedio que tiene actualmente el alumno en las asignaturas presentadas en examen ordinario o extraordinario.

Atributo numérico real en el rango 0.0 - 10.0. Para la obtención del promedio se toma la siguiente tabla de conversión para las calificaciones obtenidas, registradas en la relación *curso*:

MB	10
B	8
S	6
NA	carece de equivalente
NP	carece de equivalente
RE	carece de equivalente
AC	carece de equivalente
CO	carece de equivalente

Cardinalidad: Requerida.

***regAlumno***

Clave utilizada por DGAE para el alta de alumnos al Sistema de Registro y Control Escolar vía primer ingreso.

Atributo alfanumérico de 2 dígitos.

Cardinalidad: Requerida.

***reposición***

Atributo que indica cuantas reposiciones de credencial ha hecho el alumno.

Solamente tiene derecho a tres.

Atributo numérico de 1 dígito.

***reprobadas***

Número de asignaturas del bachillerato que han sido presentadas de alguna forma (ordinario, extraordinario, etc.); pero no han sido acreditadas.

Este atributo se obtiene sumando todas las asignaturas reprobadas registradas en la relación *curso*.

Atributo numérico de 2 dígitos.

Cardinalidad: Requerida.

***ultimoPeriodo***

Último período escolar en que el alumno presentó asignaturas del bachillerato.

Atributo alfanumérico de 3 caracteres.

***añoEsc***

Es el año escolar en el que el alumno se encuentra inscrito.

Atributo numérico de 2 dígitos.

***añoEscAnterior***

Es el año escolar anterior al que el alumno se encuentra inscrito.

Atributo numérico de 2 dígitos.

**horario**

Esta relación mantiene la información referente a los horarios y salones en los que se imparten las asignaturas correspondientes al plantel de la ENP.

**Descripción de atributos:*****cveAsig***

Clave de la asignatura.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***día***

Número de día en el que se imparte la asignatura en ese salón en ese grupo.

Atributo numérico de 1 dígito.

Cardinalidad: Requerida.

***gpo***

Grupo en el que se imparte la asignatura.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

**hora**

Este atributo indica la hora a la que se imparte la asignatura.  
Atributo de tipo fecha.

**salón**

Identificador del salón en el que se imparte la asignatura.  
Atributo alfanumérico de 6 caracteres.

**imparte**

Esta relación mantiene la información referente a los profesores que imparten una materia.  
En un grupo, por cada asignatura puede haber hasta dos profesores.

**Descripción de atributos:*****cveAsig***

Clave de la asignatura que se imparte en el grupo.  
Atributo alfanumérico de 4 dígitos.  
Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.  
Atributo numérico de 3 dígitos.  
Cardinalidad: Requerida.

***gpo***

Identificador del grupo.  
Atributo alfanumérico de 4 caracteres.  
Cardinalidad: Requerida.

***rfc***

Registro Federal de Contribuyentes del profesor.  
Atributo alfanumérico de 13 caracteres.  
Cardinalidad: Requerida.

**inscrito**

Esta relación mantiene la información de los alumnos inscritos a un grupo de examen extraordinario. El tipo de examen extraordinario puede ser EA, EB o EZ.



**Descripción de atributos:*****cveAsig***

Clave de la asignatura a presentar en el examen extraordinario.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cvePlante1***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***RPO***

Identificador del grupo de examen extraordinario.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

***noCta***

Número de cuenta del alumno que presenta el examen.

Atributo alfanumérico de 3 caracteres.

Cardinalidad: Requerida.

***periodo***

Período escolar en el que se presentó la asignatura.

Atributo alfanumérico de 3 caracteres.

Cardinalidad: Requerida.

**localiza**

Información que contiene la ubicación de los salones en los diferentes edificios del plantel.

**Descripción de atributos:*****ubicación***

Identificador del edificio o lugar de la ubicación física del salón.

Atributo numérico de 2 dígitos.

***cveEdificio***

Identificador único asignado por el sistema.

Atributo numérico entero secuencial.

Cardinalidad: Requerida.

***salón***

Identificador del salón.

Atributo alfanumérico de 6 caracteres.

Cardinalidad: Requerida.

**pertenece**

Esta relación mantiene la información de los alumnos que pertenecen a cada plantel y a su respectivo grupo.

**Descripción de atributos:*****noCta***

Número de cuenta del alumno que pertenece a la UNAM

Atributo alfanumérico de 8 caracteres.

Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel única.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***ingreso***

Año de ingreso del alumno a la ENP.

Atributo numérico de 4 dígitos.

Cardinalidad: Requerida.

***gpo***

Identificador del grupo que se asigna a los alumnos y que forma parte de una clase

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

**rectifica**

Mantiene la información de los movimientos que se realizan a las actas cuando ocurre una rectificación.

**Descripción de atributos:*****cveAsig***

Clave de la asignatura que se modifica.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***folio***

Número de folio del acta en que se asienta la calificación. Cada hoja representa un acta con diferente folio.

Atributo numérico de 7 dígitos.

Cardinalidad: Requerida.

***folioRec***

Si el acta es de rectificación, este atributo hace referencia al folio del acta que se rectifica.

Atributo numérico de 7 dígitos.

***fecha***

Fecha en la que se realiza la rectificación. Incluye día y hora.

Atributo tipo fecha

***tipo***

Este atributo indica si el acta es de rectificación o normal.

Atributo carácter, con los siguientes posibles valores:

N	acta normal.
R	acta de rectificación.

**solicitud**

Esta relación mantiene la información de los alumnos que solicitan documentos en Administración Escolar para algún propósito.

**Descripción de atributos:*****cveDocto***

Identificador del documento pedido.

Atributo numérico de 2 dígitos.

Cardinalidad: Requerida.

***noCta***

Número de cuenta del alumno que pide el documento.

Atributo alfanumérico de 8 caracteres.

Cardinalidad: Requerida.

***fechaTram***

Fecha en la que el alumno realiza el trámite.

Atributo de tipo fecha.

**solicitudMas2**

Esta relación mantiene la información de los alumnos que solicitan inscripción a más de 2 exámenes extraordinarios en un periodo de exámenes extraordinarios.

**Descripción de atributos:**

**autorizado**

Atributo que indica si el alumno tiene autorización para presentar el examen extraordinario.  
Atributo carácter, con los siguientes posibles valores:

S	si tiene autorización.
N	no tiene autorización.

Cardinalidad: Requerida.

**cveAsig**

Clave de la asignatura del grupo de examen extraordinario.  
Atributo alfanumérico de 4 dígitos.  
Cardinalidad: Requerida.

**cvePlantel**

Clave del Plantel a la que pertenece la asignatura.  
Atributo numérico de 3 dígitos.  
Cardinalidad: Requerida.

**gpo**

Identificador del grupo de examen extraordinario.  
Atributo alfanumérico de 4 caracteres.  
Cardinalidad: Requerida.

**noCta**

Número de cuenta del alumno que solicita más de dos exámenes extraordinarios.  
Atributo alfanumérico de 8 dígitos.  
Cardinalidad: Requerida.

**nomAutorizo**

Nombre de la persona que autorizó la solicitud a más de 2 exámenes extraordinarios.  
Atributo alfanumérico de 32 caracteres.

**periodo**

Período escolar en el que se presentó la asignatura.  
Atributo alfanumérico de 3 caracteres.  
Cardinalidad: Requerida.

**seriación**

Esta relación mantiene la tabla de seriación de asignaturas. Esta información es utilizada cuando se hacen las inscripciones a ordinario o extraordinario.

---

**Descripción de atributos:*****cveAsig***

Clave de la asignatura para la cual se necesita haber aprobado una asignatura anterior.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cveAsigReq***

Clave de la asignatura que se debió haber aprobado para cursar la asignatura definida por el atributo *cveAsig*.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***anioPlan***

Año a partir del cual entró en vigor el plan de estudios.

Atributo numérico de 4 dígitos.

**sinodal**

Esta relación mantiene la información de los profesores que son sinodales de algún examen extraordinario.

**Descripción de atributos:*****cveAsig***

Clave de la asignatura del grupo de examen extraordinario.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***epo***

Identificador del grupo de examen extraordinario.

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

***rfc***

Registro Federal de Causantes del profesor asignado como sinodal.

Atributo alfanumérico de 13 caracteres.

Cardinalidad: Requerida.

**tiene**

Conserva la información en las actas de las clases que se imparten en cada plantel.

**Descripción de atributos:*****cveAsig***

Clave de la asignatura de la clase correspondiente.

Atributo alfanumérico de 4 dígitos.

Cardinalidad: Requerida.

***cvePlantel***

Clave del Plantel a la que pertenece la asignatura o la clase.

Atributo numérico de 3 dígitos.

Cardinalidad: Requerida.

***entregada***

Atributo que indica si el acta ya fue entregada por el profesor o sinodal del grupo.

Atributo carácter que tiene los siguientes posibles valores:

N no ha sido entregada.

S ya fue entregada.

***folio***

Número de folio del acta en que se asienta la calificación. Cada hoja representa un acta con diferente folio.

Atributo numérico de 7 dígitos.

Cardinalidad: Requerida.

***tipo***

Este atributo indica si el acta es de rectificación o normal.

Atributo carácter, con los siguientes posibles valores:

N acta normal.

R acta de rectificación.

***gro***

Grupo en donde se imparte la asignatura

Atributo alfanumérico de 4 caracteres.

Cardinalidad: Requerida.

## VI.5 DIAGRAMAS DE TRANSICIÓN DE ESTADOS

Un diagrama de transición de estado es un modelo utilizado cuando ocurren eventos en un proceso o como consecuencia de la interacción con el usuario.

Un diagrama de transición de estado está compuesto de un estado, una transición, una condición de transición y una acción de transición.

Se dice que un estado es pasivo por que representa la espera de que ocurra algo. Un diagrama de transición de estado sólo puede representar un estado al mismo tiempo.

El nombre del estado representa qué es lo que el proceso está esperando, qué es lo que el proceso está realizando o qué será desplegado en pantallas.

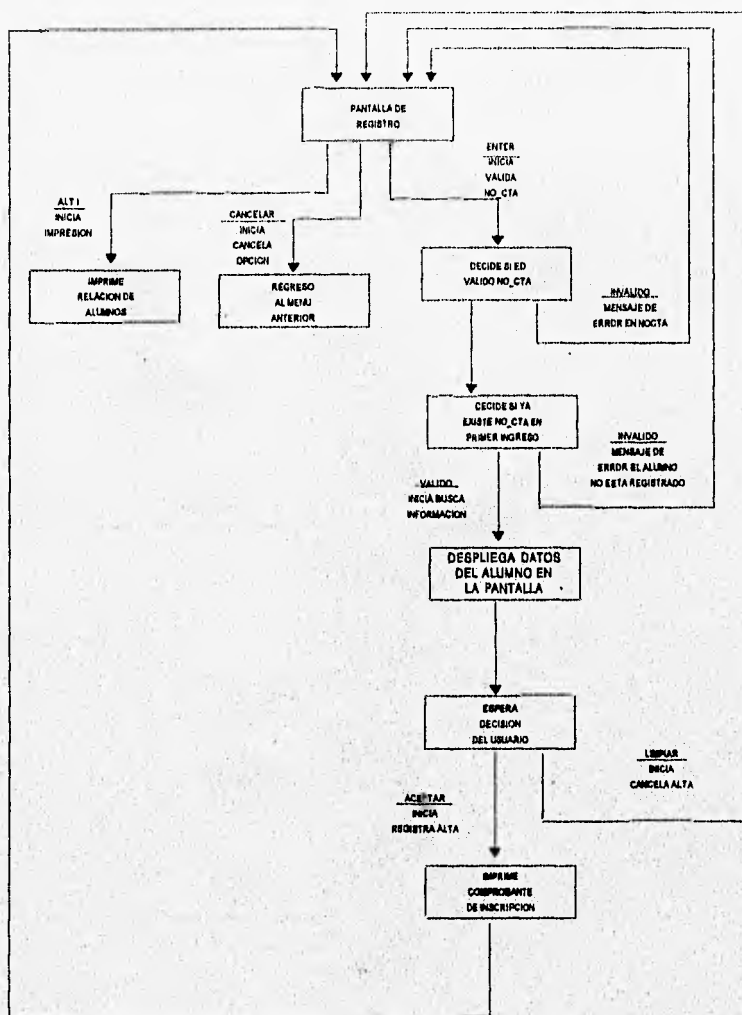
Un estado inicial es denotado por una transacción de entrada y describe lo que está ocurriendo antes de que se ejecute cualquier condición. Un estado final no tiene transiciones de salida y describe una situación, la cual no es posible cambiar.

Una transición representa un cambio de estado y está formada por la condición de transición y por la acción de transición.

Múltiples transacciones de o a un estado son permitidas y también es posible que una transacción regrese al mismo estado del cual se originó. Una condición de transacción indica qué es lo que ocasionó que el sistema cambiara de un estado a otro. Una acción de transacción ocurre cuando se cumple la condición. La acción puede causar procesos o señales que deberán ser ejecutados.

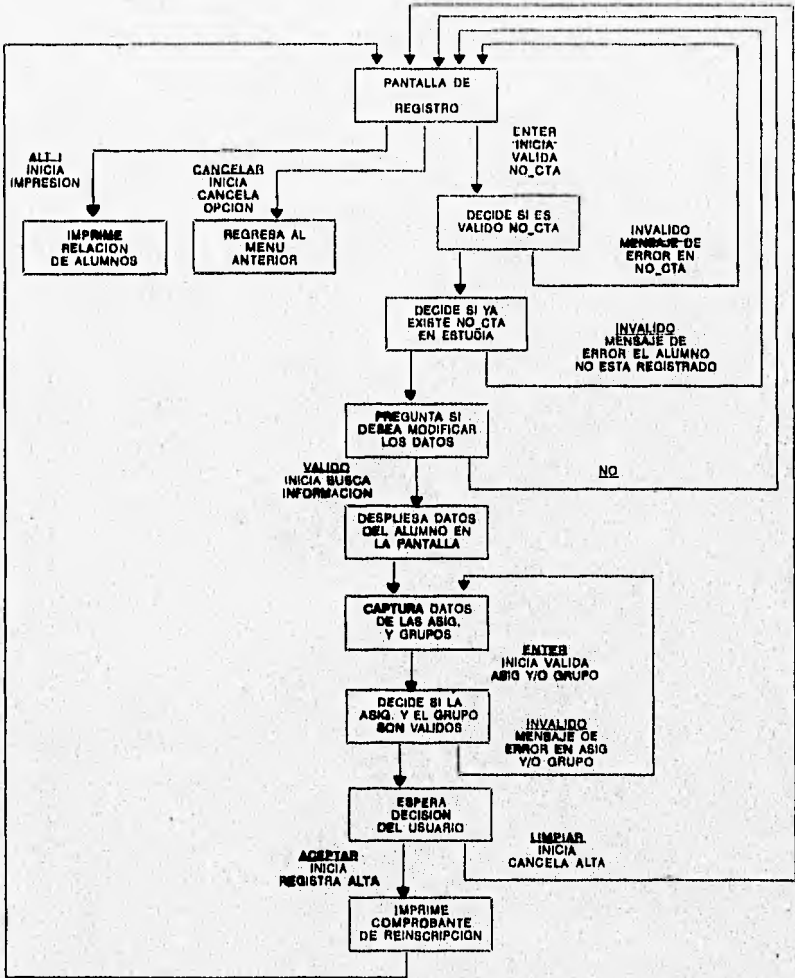
A continuación mostramos los diagramas de transición de estado del sistema.

CONTROLA ALUMNOS INSCRITOS

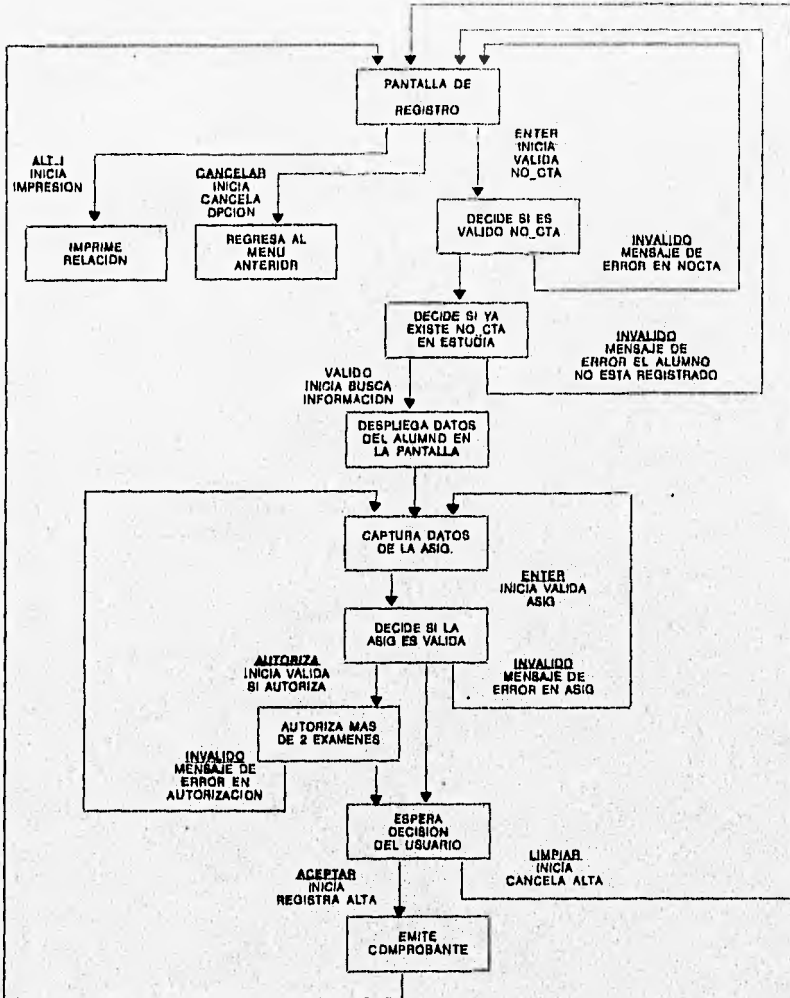




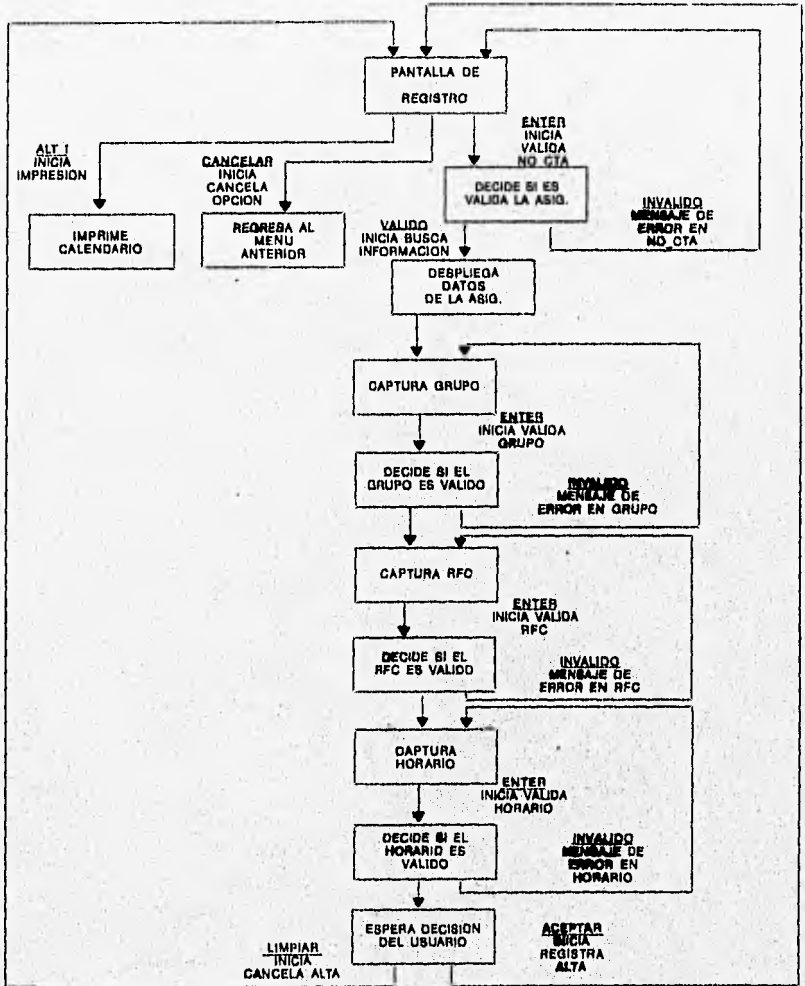
**CONTROLA ALUMNOS REINSCRITOS**



### EXÁMENES EXTRAORDINARIOS



CALENDARIO DE EXÁMENES EXTRAORDINARIOS



## VI.6 DIAGRAMAS DE ESTRUCTURA

Son una herramienta que sirve para representar en forma gráfica las partes componentes de un proceso del sistema. Estas partes representadas son:

- Módulos componentes del proceso
- Conexiones entre los módulos
- Jerarquía de los módulos

Un módulo es un grupo de instrucciones tratadas como una unidad y es invocado por su nombre desde otras unidades y que a su vez puede llamar a otros módulos.

El proceso se organiza en módulos donde cada uno de éstos realice una función perfectamente bien definida y sea lo suficientemente pequeño para entenderse sin dificultades; estas características contribuyen a facilitar la implantación y el mantenimiento del sistema.

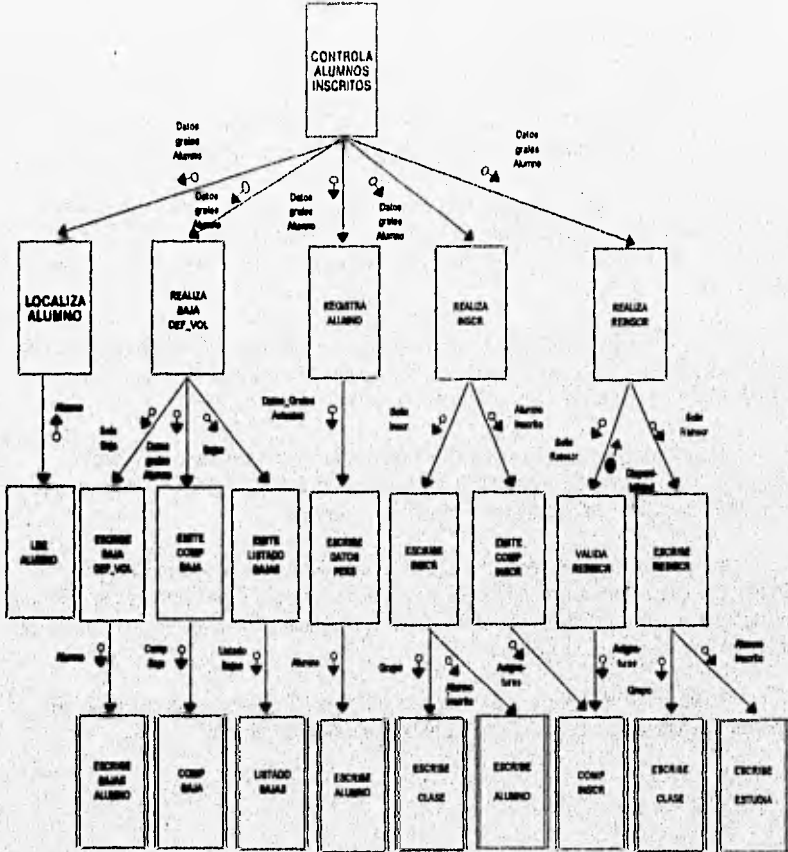
Para una buena organización de los módulos, el diseñador debe tener en mente una clara separación de lo que es el control y el trabajo en sí del proceso; asimismo, debe mantener una organización descendente de las funciones del proceso.

En todo diagrama de estructura los módulos se encuentran unidos por flechas, mismas que representan una llamada entre módulos y en cada llamada se puede presentar un paso de parámetros, ya sea de datos o de señales de control.

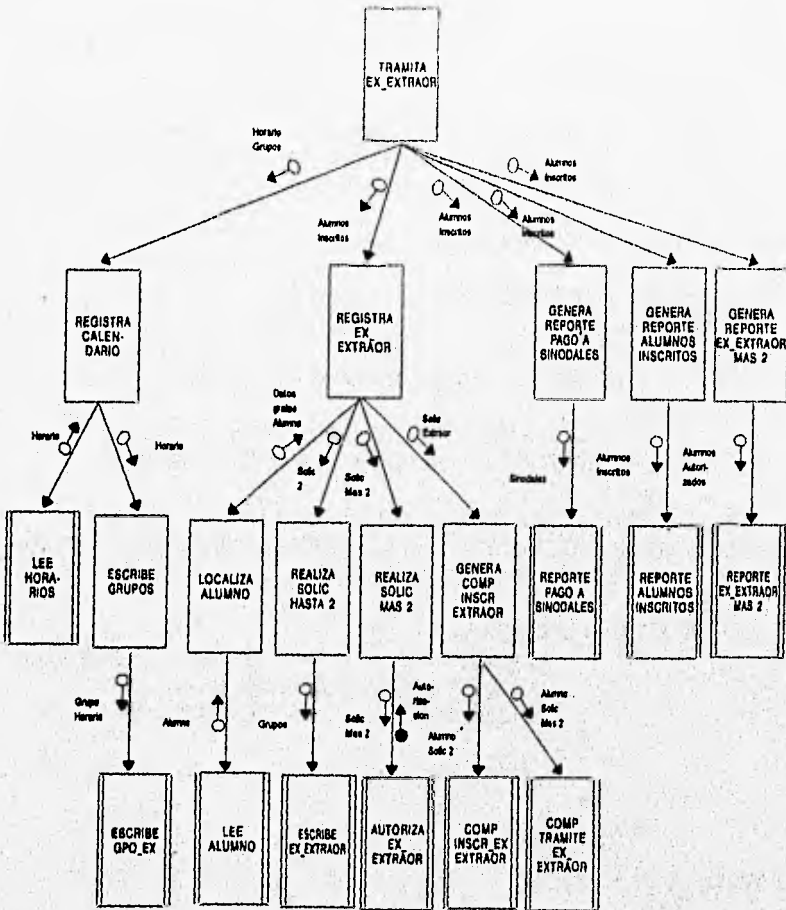
Los parámetros se representan con un círculo del cual se desprende una pequeña flecha. Si el círculo está vacío indica que el parámetro es uno o varios datos, de lo contrario se tratará de un parámetro de control. Finalmente, la pequeña flecha apunta del módulo que envía el parámetro hacia el que lo recibe.

A manera de ejemplo, a continuación se muestran los diagramas de estructura para el Control de alumnos inscritos y el trámite para examen extraordinario.

CONTROLA ALUMNOS INSCRITOS



### TRAMITA EX\_EXTRAO



**VI.6.1 Especificación de Módulos**

Se trata de una explicación que se elabora para cada módulo de un diagrama de estructura.

“Una especificación de módulo detallará las acciones requeridas en cada módulo en términos técnicos”

Estas explicaciones técnicas posteriormente se traducirán en código del programa final, por lo que el diseñador debe tomar en cuenta que el código del sistema debe ser flexible, de fácil mantenimiento, reusable e implantable.

**1.1 CONTROLA ALUMNOS INSCRITOS****LOCALIZA ALUMNO(IDENTIFICA ALUMNO)**

Si inscripción

Si primer Ingreso

INGRESA INSCRIPCIÓN(GRALES\_ALUMNO)

Otro

VALIDA REINSCRIPCIÓN(GRALES\_ALUMNO)

REGISTRA REINSCRIPCIÓN(GRALES\_ALUMNO)

Fin si

**GENERA COMP\_INSCRIP(SOLIC\_COMP\_INSCR)****GENERA RELACIÓN ALUMNOS INSCRITOS(SOLIC\_REL\_ALUMNO\_REINSCR)**

Otro

Si baja Definitiva

REGISTRA BAJA\_DEF\_VOL(GRALES\_ALUMNO)

GENERA COMP\_BAJA(DATOS\_BAJA)

GENERA LISTADO\_BAJA(SOLC\_REL\_BAJAS)

Otro

REGISTRA ALUMNO(DATOS\_PERS)

Fin Si

Fin Si

**2 TRAMITA EX\_EXTRAOR****ESCRIBE CALENDARIO(CALENDARIO)****LOCALIZA ALUMNO(IDENTIFICAALUMNO)****ESCRIBE EX\_EXTRAOR(GRALES\_ALUMNO)**

Si más de 2 asignaturas

**AUTORIZA EX\_EXTRAOR(PERMISO\_EXTRAOR\_AUTOR)**

Fin Si

**GENERA COM\_INSCR\_EX\_EXTRAOR(DATOS\_EXTRAOR)**

---

GENERA REPORTE PAGO A SINODALES(SOLIC\_REP\_PAGO\_SINOD  
\_EXTRAOR)  
GENERA REPORTE ALUMNOS INSCRITOS(SOLIC\_REP\_ALUMNO\_INSC  
\_ASIG\_JURADO)  
GENERA REPORTE EX\_EXTRAOR MAS 2(SOLIC\_REP\_ALUMNO\_AUTO  
\_MAS\_2)

### **VI.7 ARQUITECTURA DE CONEXIONES**

La arquitectura de conexiones se refiere a la forma en que se conectarán los equipos de la DGENP de la UNAM con los 9 planteles de la ENP y otras entidades externas. Estas conexiones están basadas en protocolos de comunicación que en su momento se detallaron.

El propósito de la comunicación de la DGENP con sus entidades, es contar con información en línea disponible en cualquier momento, de ésta manera se agilizarán trámites y se ahorrarán recursos.

En la figura VI.7.1 se esquematiza la forma de conexión entre la DGENP y planteles.

En la figura VI.7.2 se esquematiza la conexión interna de los planteles.



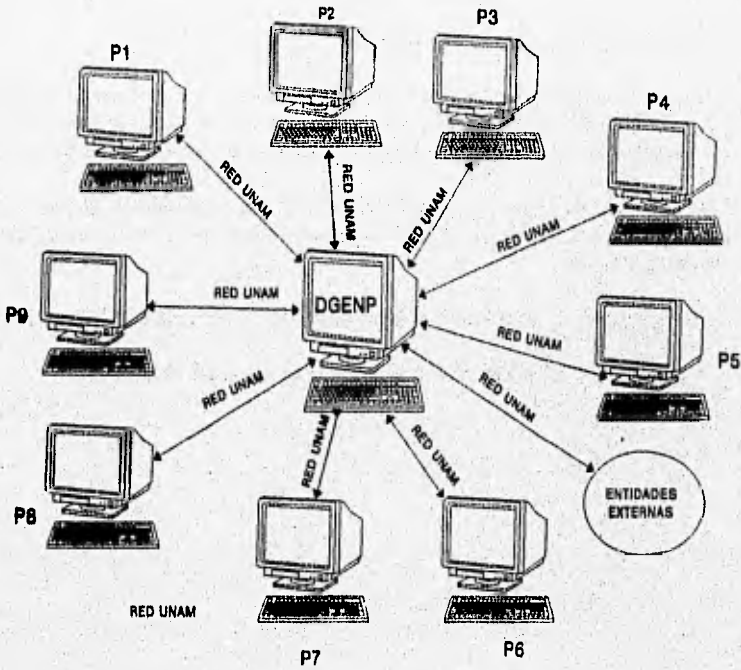


Fig. VI.7.1 Conexión entre DGENP y planteles





## CONCLUSIONES

La implantación de este sistema permitirá a la Unidad de Administración Escolar realizar de una manera más fácil y rápida sus actividades lo que se traduce en un mejor servicio para la comunidad académica y estudiantil.

El nuevo sistema permitirá el ahorro de tiempo y recursos facilitando los trámites escolares.

El contar con un registro de todos los trámites que se llevan a cabo en la Unidad de Administración Escolar permitirá la emisión de documentos oficiales y estadísticas en forma automática apoyando a las autoridades en la toma de decisiones.

El sistema sentará las bases para lograr una automatización integral permitiendo más adelante que el mismo alumno realice sus trámites desde terminales que se encuentren en su escuela, o bien, vía telefónica inclusive desde su misma casa.

En una primera instancia el costo del sistema puede ser elevado debido a la tecnología bajo la cual está desarrollado pero garantiza que no se volverá obsoleto tan rápidamente y facilita la realización de un sistema integral en un futuro gracias a que es una arquitectura abierta.

Como ya se mencionó anteriormente el sistema está desarrollado bajo una arquitectura cliente/servidor que reduce el tráfico en la red debido a que se distribuye la carga de trabajo entre el cliente y el servidor, lo que trae consigo un mejor tiempo de respuesta.

En cada uno de los planteles ya se cuenta con equipo, que se aprovechará para el sistema y además para otras tareas que se tengan asignadas en el área de la administración. De esta forma el equipo se utiliza y se aprovechan al máximo los recursos.

No se niega que se han desarrollado diversos sistemas de administración escolar, la mayoría en plataformas de PC's, que incluyen los módulos principales como son inscripciones, reinscripciones y exámenes extraordinarios. Sin embargo, la propuesta presentada en este trabajo requirió de un análisis y diseño integral más detallado debido a la complejidad que tienen los procesos que se realizan en la Escuela Nacional Preparatoria y además tomando en cuenta el caso de la ENP plantel no.2 "Erasmus Castellanos Quinto", que cuenta con iniciación universitaria (secundaria).



---

**BIBLIOGRAFÍA**

---

- [BYTE]     BYTE , edición en español.  
          Coorporación Editorial S.A. de C.V.  
          México, ed. junio de 1995
- [COA90]    Coad, P., y E. Yourdon  
          "Object-Oriented Analysis"  
          Prentice-Hall  
          California, 1990.
- [DAT86]    Date, C.J.  
          "Introducción a los Sistemas de Bases de Datos"  
          Addison Wesley Iberoamericana  
          tercera edición  
          Wilmington, 1986
- [DEW93]    Dewire, Dawna Travis  
          "Client/Server Computing"  
          Mc Graw-Hill  
          New York, 1993
- [ELB94]    Elbert, Bruce y Martyna, Bobby  
          "Client/Server Computing Architecture, Applications and Sitributed  
          Systems Management"  
          Artech House  
          Boston, 1994
- [HAN92]    Management and Design"  
          Prentice-Hall  
          Englewood Cliffs, 1992
- [KOR86]    Korth, Henry F. y Silberschatz, Abraham  
          "Data Base System Concepts"  
          Mc Graw Hill Advanced Computer Science Series  
          New York, 1986

- 
- [MCG93] McGoveran, D. y Date, C.J.  
"A Guide to Sybase and SQL Server"  
Addison-Wesley  
New York, 1993
- [ORF94] Orfali, Robert y Harkey, Dan  
"Essential Client/Server Survival Guide"  
Van Nostrand Reinhold  
New York, 1994
- [PC MONITOR] PCMONITOR  
Editorial NESS S.A. de C.V.  
México, ed. junio de 1995
- [PRE93] Pressman, Roger S.  
"Ingeniería del Software un Enfoque Práctico"  
Mc Graw Hill  
Tercer Edición  
Madrid, 1993
- [RED] RED  
Editorial RED S.A. de C.V.  
números 338,343,355  
México, 1995.
- [SAL93] Salemi, Joe  
"Guide to Client/Server Databases"  
Ziff-Davis Press  
Emeryville, 1983
- [TOW92] Townsend, James J.  
"Introduction to Databases"  
QUE  
Carmel, 1992

[YOU93] Yourdon, Edward  
"Análisis Estructurado Moderno"  
Prentice Hall  
México, 1993

[VAR94] Varios  
"Data Based Advisor"  
Noviembre 1994